



AFOSR-TR 91 0741

(2)

Fast Algorithms for Linear Least-Squares
estimation of Multi-Dimensional Random Fields

Andrew E. Yagle

FINAL TECHNICAL REPORT

Grant # AFOSR 89-0017

DTIC
ELECTE
SEP 09 1991
S B D

Approved for public release;
distribution unlimited.



DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

THE UNIVERSITY OF MICHIGAN

Department of Electrical Engineering
and Computer Science

Ann Arbor, Michigan 48109-2122
USA

91-09736



OFFICE OF SCIENTIFIC RESEARCH & ASSOCIATED
TECHNOLOGICAL TO DTIC
This report has been reviewed and
approved for release in accordance with
the provisions of the unlimited,
classification of the report.
STINFO Program Manager

Air Force Office of Scientific Research

**Fast Algorithms for Linear Least-Squares
Estimation of Multi-Dimensional Random Fields**

Andrew E. Yagle

FINAL TECHNICAL REPORT

Grant # AFOSR 89-0017

Principal Investigator:

Andrew E. Yagle, Assistant Professor

Dept. of Electrical Engineering and Computer Science

The University of Michigan, Ann Arbor, MI 48109-2122

Phone: (313) 763-9810. Fax: (313) 763-1503. E-mail: aey@dip.eecs.umich.edu

Date of This Report: July 1991.

Directorate of Mathematical Sciences


REPORT DOCUMENTATION PAGE

1. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			16. RESTRICTIVE MARKINGS		
2. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited		
7. DECLASSIFICATION/DOWNGRADING SCHEDULE			5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-89-0017		
PERFORMING ORGANIZATION REPORT NUMBER(S)			9. MONITORING ORGANIZATION REPORT NUMBER(S)		
4. NAME OF PERFORMING ORGANIZATION University of Michigan		5b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research		
6. ADDRESS (City, State and ZIP Code) 1301 Beal St. Ann Arbor, MI 48109-2122			7b. ADDRESS (City, State and ZIP Code) Bolling Air Force Base Washington, DC 20332-6448		
8. NAME OF FUNDING/SPONSORING ORGANIZATION		9b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-89-0017		
10. ADDRESS (City, State and ZIP Code)			10. SOURCE OF FUNDING NOS.		
			PROGRAM ELEMENT NO. 61102 F	PROJECT NO. 2304	TASK NO. A6
11. TITLE (Include Security Classification) Fast Algorithms for Linear Least-Squares Estimation of Multi-Dimensional Random Fields					
12. PERSONAL AUTHOR(S) Dr. Andrew E. Yagle					
13. TYPE OF REPORT Final Technical		13b. TIME COVERED FROM 11/1/88-5/31/91	14. DATE OF REPORT (Yr., Mo., Day) July 31, 1991		15. PAGE COUNT
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB. GR.	Random fields, fast algorithms, linear least-squares estimation, Levinson algorithm, linear prediction, spectral estimation.		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>This report develops fast algorithms for computing filters for linear least-squares estimation of one, two, and three-dimensional random fields. The algorithms generalize the split Levinson and Schur algorithms to two and three dimensions; however, they are applicable to a more general Toeplitz-plus-Hankel structure in the covariance function. A discrete version of the Bellman-Siegert-Krein resolvent identity is developed for smoothing problems in one and two dimensions. Applications to linear predictive coding, and restoration and smoothing, of isotropic random fields on a polar raster are demonstrated. In addition, two new algorithms are developed for spectral estimation on a two-dimensional polar raster. Both use the Radon transform to map the 2-D problem into 1-D problems. Interpolating functions for computing the Radon transform, positive definite covariance extensions, and correlation matching are all considered.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT CLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Jon Sjoogren			22b. TELEPHONE NUMBER (Include Area Code) 202-767-4940	22c. OFFICE SYMBOL NM	

TABLE OF CONTENTS

1. Quick Summary of Research Accomplishments	1
2. Quick Review of Linear Prediction Fast Algorithms	
2.1 One-Dimensional Levinson, Schur, and Split Algorithms	2
2.2 Two-Dimensional Levinson and Schur Algorithms	3
3. Research Objectives	4
4. Research Accomplishments	
I. Fast Algorithms for Optimal Filters	5
II. Algorithms for Covariance and Spectral Estimation	10
5. References	13
6. Publications Supported by This Grant	14

APPENDICES A-J



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

1. QUICK SUMMARY OF RESEARCH ACCOMPLISHMENTS

1. A new fast algorithm for solving Toeplitz-plus-Hankel systems of equations. The new algorithm appears to be 33% faster than the previous approach of reformulating the problem as a block-Toeplitz system of equations.
2. A new fast algorithm for solving *block*-Toeplitz-plus-Hankel systems of equations. This algorithm is useful for linear prediction for two-dimensional random fields defined on a discrete polar raster. The covariance must be a Toeplitz-plus-Hankel function of both the radial and angular arguments; an isotropic random field has this property.
3. A fast algorithm for linear prediction for three-dimensional random fields defined on a spherical raster. The covariance must be a Toeplitz-plus-Hankel function of radius and of the two angular arguments; a time-varying random field that is wide-sense stationary in time has this property.
4. A discrete form of the Bellman-Siebert-Krein resolvent identity, which can be used to compute smoothing filters from the prediction filters computed using the algorithms in #2 and #3 above. This generalizes a one-dimensional (1-D) continuous-parameter result of Kailath to: (1) the discrete case; and (2) two dimensions (2-D).
5. Two new algorithms for estimating a structured Toeplitz-plus-Hankel covariance function from time series data in 1-D or 2-D. The estimated covariances have the structure required by the algorithms in #2 and #3 above.
6. The two-dimensional linear prediction problem on a 2-D polar raster. Includes: (1) two new algorithms for spectral estimation, using Radon transforms to map the 2-D problem into 1-D problems; (2) interpolating functions to compute Radon transforms; and (3) positive-definite covariance extension and correlation matching.
7. Some proposed VLSI implementations of the 1-D and 2-D algorithms described above. The similarity of these algorithms to finite-difference equations allows VLSI for finite-difference equations to be adapted to these algorithms, with some changes.
8. Demonstrations of the new algorithms applied to the problems of: (1) linear predictive coding of images defined on a polar raster; and (2) smoothing and restoration of these images. Such images arise in tomography and spotlight synthetic aperture radar.

2. QUICK REVIEW OF LINEAR PREDICTION FAST ALGORITHMS

Linear least-squares estimation has played an important and useful role in modern signal processing. It has been applied to problems in one-dimensional prediction and estimation with considerable success. In roughly the last decade, similar success has been achieved for multidimensional estimation and smoothing problems.

In order to place the results of this report in proper perspective, it is worthwhile to briefly review some fast algorithms used in linear least squares estimation. More details on this material are available in Section 2 of Appendix A.

2.1 One-Dimensional Levinson, Schur, and Split Algorithms

In the one-dimensional case, for a wide-sense stationary random process, the linear prediction problem can be solved efficiently using the celebrated Levinson algorithm [1]. This algorithm utilizes the Toeplitz structure of the covariance matrix to reduce the number of multiplications required to solve the N th order prediction problem from the $O(N^3)$ required by Gaussian elimination to $O(N^2)$. The Levinson algorithm recursively computes the prediction filters in increasing order. In the process, it generates a set of reflection coefficients that constitute an alternative parametrization of the prediction filters.

In the Levinson algorithm, the reflection coefficients must be computed using an "inner product" expression (equation (2-1b) of Appendix A), which accounts for roughly one-third of the computation in the algorithm. More importantly, this computation is not parallelizable. The "inner product" computation can be avoided by using the Schur algorithm [2] to compute the reflection coefficients directly from the covariance of the random process. Thus a more efficient procedure for computing the linear prediction filters is to run the Schur algorithm in parallel with the Levinson algorithm, using the reflection coefficients computed by the Schur algorithm in the Levinson algorithm [3].

Recently Delsarte and Genin [4] noted that a redundancy exists in the lattice computations in the Levinson and Schur algorithms. By replacing the lattice recursions with a single three-term recurrence, half of the multiplications in the lattice recursions are avoided. This results in the *split* Levinson and Schur algorithms, which are obviously more

efficient implementations of the classical Levinson and Schur algorithms.

In the split algorithms, the reflection coefficients parametrizing the prediction filters are replaced by “potentials” that also parametrize these filters. The split Schur algorithm can be used to compute these potentials from the covariance function; the potentials are then inserted into a split Levinson algorithms running in parallel. More importantly, the split algorithms are the axis along which the one-dimensional Levinson and Schur algorithms can be extended to higher dimensions, and to more general covariance structures.

2.2 Two-Dimensional Levinson and Schur Algorithms

There have been several efforts to generalize the Levinson and Schur algorithms to two dimensions, in order to simplify the solution of the two dimensional linear prediction problem. We quickly summarize these here; for more details see Appendices A and C.

The usual approach is to assume that a two-dimensional random field is [5]-[7]: (1) defined on a rectangular array of points; (2) stationary; (3) has quarter-plane or asymmetric half-plane causality, i.e., the linear prediction filter for the random field should have quarter-plane or asymmetric half-plane support. Then the two-dimensional linear prediction problem can be formulated as a *multichannel one-dimensional* problem, and solved using the multichannel Levinson and Schur algorithms [8].

The multichannel Levinson and Schur algorithms are essentially matrix versions of the one-dimensional algorithms, and they exploit the Toeplitz-block-Toeplitz structure of the covariance matrix to similarly reduce the number of multiplications needed to solve the two-dimensional discrete Wiener-Hopf (or Yule-Walker) equations. There are several variations on this theme, but all essentially reformulate the two-dimensional problem on a rectangular lattice as a multichannel one-dimensional problem of some kind.

3. RESEARCH OBJECTIVES

The goals of this project were as follows:

1. To develop two-dimensional versions of the Levinson and Schur algorithms that relax the causality requirements of existing two-dimensional algorithms, and replace them with causality assumptions that are more physically reasonable. These algorithms should also not require stationarity of the random field, but allow a more general structure in the covariance function;
 2. To develop algorithms for the *smoothing* problem, as opposed to the prediction problem, for random fields. Since random fields are in general not causally generated, the use of the *prediction* filters computed using the algorithms in #1 is limited to linear predictive coding of the random field. Linear least squares filters suitable for reducing noise and restoration should be smoothing filters that use all the noisy data to estimate the random field at any point;
 3. To develop *three*-dimensional versions of the algorithms in items #1 and #2, suitable for three-dimensional random fields. Such random fields describe random processes defined over space, e.g., temperature, images varying in time, etc.;
 4. To successfully implement these algorithms, study their numerical behavior, and apply them to some problems in image restoration, smoothing, and linear predictive coding.
- All four goals have been successfully accomplished, as this report will demonstrate.
- In addition, we have accomplished the following additional goals:
5. To develop algorithms for estimating from 1-D and 2-D time series data covariances with the structure required by the above algorithms;
 6. To study the two-dimensional linear prediction problem on a polar raster, and develop two-dimensional spectral estimation algorithms that ensure non-negative spectral estimates;
 7. To develop possible VLSI implementations of the generalized Levinson and Schur algorithms.

4. RESEARCH ACCOMPLISHMENTS

This section contains a concise summary of our research results. Technical details are provided in the Appendices, as noted below. Part I includes Sections 4.1-4.5, and covers development of fast algorithms for determining optimal filters for least-squares estimation of random fields. Part II includes Sections 4.6-4.9, and covers estimation of structured covariances from 1-D and 2-D time series data, spectral estimation on a polar raster from 2-D time series data, applications, and VLSI implementations.

All of the results presented below are new contributions to the field of linear prediction.

Part I: Fast Algorithms for Optimal Filters

4.1 Continuous-Parameter Results

Our original proposal was formulated in continuous-parameter space, since our preliminary results were all continuous-parameter algorithms. Specifically, the goal was to develop fast algorithms for solving the multi-dimensional Wiener-Hopf integral equation

$$k(x, y) = h(x, y) + \int_{|z| \leq |x|} h(x, z)k(z, y)dz, \quad |y| \leq |x|, \quad x, y \in R^n, n = 1, 2, 3$$

The solution $h(x, y)$ of this integral equation is the optimal linear least-squares filter for computing the estimate $\hat{s}(x)$ of a zero-mean random field with covariance $k(x, y)$ from noisy observations $\{w(z) = s(z) + v(z), |z| \leq |x|\}$, where $v(z)$ is zero-mean white noise.

In [9] and [10] we noted for $n = 3$ that if the covariance function $k(x, y)$ satisfies $(\Delta_x - \Delta_y)k(x, y) = 0$, then the prediction filter $h(x, y)$ satisfies the differential form

$$(\Delta_x - \Delta_y)h(x, y) = \int_S V(x, e)h(|x|e, y)|x|^2 de; \quad V(x, e) = -\frac{2}{|x|^2} \frac{d}{d|x|} |x|^2 h(x, |x|e)$$

where S is the unit sphere and e is a unit vector in R^3 .

The derivation of this equation and its implications are discussed extensively in Appendix A. Here we merely note some significant facts:

1. The differential form can be viewed as some sort of three-dimensional, continuous-parameter generalization of the split Levinson recurrence. $V(x, e)$ is a similar generalization of the potential in the split algorithms;

2. A similar differential form, initialized using the covariance function $k(x, y)$, can be used to compute $V(x, e)$ from $k(x, y)$. This can be viewed as a three-dimensional, continuous-parameter generalization of the split Schur recurrence;
3. The differential form can be propagated recursively in increasing prediction "order" $|x|$, for all $|y| < |x|$. In this way, it is possible to solve the three-dimensional Wiener-Hopf equation recursively in increasing $|x|$;
4. The structure $(\Delta_x - \Delta_y)k(x, y) = 0$ required in the covariance function can be viewed as a generalization of the block-Toeplitz structure required by previous two-dimensional Levinson algorithms. However, it is much more general: note that isotropic ($k(x, y) = k(|x - y|)$) and homogeneous ($k(x, y) = k(x - y)$) random fields are included as special cases of this structure;
5. The causality assumed in the random field prediction filters $h(x, y)$ is simply that $h(x, y) = 0$ for $|y| > |x|$, i.e., causality is defined simply in terms of radius. This is more reasonable physically than quarter-plane causality, lexicographic ordering, etc.

The above differential forms generate the prediction filter for the random field. However, for estimation, noise reduction, and image restoration, the *smoothing* filter which uses *all* noisy observations (including $|y| > |x|$) is desirable. In the one-dimensional case, Kailath [10] has shown that the smoothing filter can be easily obtained from the prediction filter using the Bellman-Siebert-Krein resolvent identity. For our purposes, this is simply a differential equation relating the smoothing and prediction filters. A three-dimensional generalization of the result of [10], applicable to the smoothing problem for three-dimensional random fields, is derived in Appendix A, which consists of the following paper: A.E. Yagle, "Analogues of Split Levinson, Schur, and Lattice Algorithms for Three-Dimensional Random Field Estimation Problems," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1780-1799, Dec. 1990.

A major part of this project has focused on deriving inherently discrete versions or counterparts to the above continuous algorithms, and one and two dimensional versions of the above three-dimensional algorithms. There are several reasons for this:

1. Since the data to be processed is most likely sampled or discrete in nature, the actual problem of interest is the discrete version;

2. Any continuous algorithms must ultimately be discretized before they can be implemented on a computer; however, discretization errors will be eliminated if an inherently discrete version of the algorithms, applicable to discrete problems, can be found;
3. The one-dimensional and two-dimensional cases are important in their own right. The two-dimensional case is particularly important, due to image processing applications;
4. Our initial attempts along these lines quickly met with success (see below).

4.2 One-Dimensional Toeplitz-Plus-Hankel Systems

The one-dimensional discrete version of this algorithm is quite interesting in its own right. It is a generalization of the split Levinson and Schur algorithms that solves Toeplitz-plus-Hankel systems of equations (i.e., systems of equations in which the system matrix is the sum of an arbitrary Toeplitz matrix and an arbitrary Hankel matrix). This algorithm requires only half as many multiplications as a previous algorithm [11] for such systems of equations.

Toeplitz-plus-Hankel systems of equations arise in linear-phase prediction filter design, the Hildebrand-Prony spectral line estimation procedure, PADE approximation, and atmospheric scattering, in addition to the nonstationary process linear prediction application motivating this algorithm here.

The heart of the algorithm is a *four*-term recurrence that uses *two* potentials, as compared to the usual split algorithm recurrence that is a three-term recurrence using a single potential. Since a Toeplitz-plus-Hankel system has twice as many degrees of freedom as the purely Toeplitz system solved by the usual split algorithms, this is reasonable, and it seems to be efficient.

Details are given in Appendix B, which consists of the paper: A.E. Yagle, "New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-39, no. 11, Nov. 1991. These include application to Toeplitz-plus-Hankel normal and Yule-Walker equations, arbitrary Toeplitz-plus-Hankel systems of equations, and simplifications to the classical split algorithms [4] for purely Toeplitz systems.

4.3 Two-Dimensional Block-Toeplitz-Plus-Hankel Systems

The two-dimensional discrete version of this algorithm solves *block* Toeplitz-plus-Hankel systems of equations. This algorithm is useful for linear prediction for two-dimensional random fields defined on a *discrete polar raster*. The covariance must be a Toeplitz-plus Hankel function of both the radial and angular arguments; the important case of an *isotropic* random field has this property.

Random fields defined on a discrete polar raster arise in tomography and spotlight synthetic aperture radar. Although such data could be interpolated onto a rectangular lattice, this is necessarily inexact; it also affects the covariance function. For example, the covariance function for an isotropic random field on a rectangular lattice is a Toeplitz function of both the abscissae and the ordinates, leading to a Toeplitz-block-Toeplitz covariance matrix in the two-dimensional discrete Wiener-Hopf equation. The multichannel Levinson algorithm can be used on this system.

However, the covariance function for an isotropic random field on a polar raster is a Toeplitz-plus-Hankel function of the radii and a Toeplitz function of the angular arguments, leading to a block Toeplitz-plus-Hankel covariance matrix in the two-dimensional discrete Wiener-Hopf equation. The multichannel Levinson algorithm *cannot* be used to solve this problem—only the new algorithm of this section is applicable.

Remarkably, the basic recurrence for this algorithm is essentially a discrete version of the continuous-parameter differential form, with the Laplacians becoming discrete Laplacians and the integral becoming a sum. This is remarkable since the explicitly discrete algorithm is an *exact* solution to the discrete problem, rather than just a discretized form of the continuous algorithm. In the continuous limit, the discrete algorithm approaches the continuous differential form, as expected.

Details are given in Appendix C, which consists of the paper: W.-H. Fang and A.E. Yagle, "Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992. This includes a discussion of application to isotropic and other random fields, details of the reduction to the continuous case, and resulting simplifications.

4.4 Three-Dimensional Block-Toeplitz-Plus-Hankel Systems

The three-dimensional discrete version of this algorithm solves the linear prediction problem for three-dimensional random fields defined on a spherical raster. The covariance must be a Toeplitz-plus-Hankel function of radius and of the two angular arguments; a time-varying random field that is wide-sense stationary in time has this property.

This result is a direct extension of the 2-D algorithm. For a summary and derivation of this algorithm, see Appendix D.

4.5 One and Two-Dimensional Discrete Bellman-Siebert-Krein (BSK) Resolvent Identities

Kailath [10] has noted the applicability of the BSK resolvent identity to computing one-dimensional smoothing filters from prediction filters. We have developed a discrete version of Kailath's result, and numerically implemented it. We have also developed a two-dimensional discrete version of the BSK relating the prediction filters for two-dimensional random fields on a polar raster to the smoothing filters for such random fields. The two-dimensional discrete algorithm has also been successfully implemented numerically.

The significance of this result is noted in #3.6 below, in which the improvement in using smoothing filters instead of prediction filters is demonstrated on several examples. For a polar raster with N points along each of N radial directions, the number of multiplications needed to compute the smoothing filter is reduced from $O(N^6)$ using Gaussian elimination to $O(N^4)$, if the algorithm in #3.4 is used to compute the prediction filters and the discrete BSK algorithm is then used to compute the smoothing filters.

Details are given in Appendix E, which consists of the paper: W.-H. Fang and A.E. Yagle, "Fast Algorithms for Linear Least-Squares Smoothing Problems in One and Two Dimensions using Generalized Discrete Bellman-Siebert-Krein Resolvent Identities," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992. It includes details of the reduction to the continuous case, and resulting simplifications. The continuous case is treated in [10] for the one-dimensional case, and in Section 5 of Appendix A for the three-dimensional case.

Part II: Algorithms for Covariance and Spectral Estimation

4.6 Structured Estimation of Covariances

This second part covers research into estimating an unknown covariance, with the (block) Toeplitz-plus-Hankel structure required by the above algorithms, from 1-D or 2-D time series data. In this section we discuss covariance estimation; in the next, spectral estimation.

There has been much work on this problem for estimating *stationary* covariance functions from data. A common procedure is to estimate autocorrelation lags from the data, form a covariance matrix, and then "Toeplitzify" it by averaging along the diagonals of the covariance matrix. This procedure projects (defined from the Hilbert-Schmidt inner product) the data lag matrix onto the subspace of symmetric Toeplitz matrices.

We have extended this approach. We have derived an algorithm that projects the data lag matrix on the subspace of symmetric Toeplitz-plus-Hankel matrices. This subspace is computed using a Gram-Schmidt orthonormalization. The procedure finds the closest (Hilbert-Schmidt norm) symmetric Toeplitz-plus-Hankel matrix to the given data lag matrix. Unfortunately, this procedure is more complicated than simply averaging along diagonals. The Toeplitz projection is found this way; however, the Hankel part of the projection requires weighted sums of some data lag matrix elements.

Due to the complexity of this algorithm, we have developed a second algorithm that truly generalizes the "Toeplitzation" of averaging along diagonals into a "Toeplitz-plus-Hankelization" of averaging along diagonals and antidiagonals. The resulting estimated Toeplitz-plus-Hankel matrix has slightly more structure than required, but the algorithm is much simpler than the first algorithm. In addition, constraints such as positive definiteness and rank constraints can be incorporated into a slightly different but equivalent form of this algorithm. Finally, a two-dimensional version of this latter algorithm has also been derived.

Details are given in Appendix F, which consists of the paper: W.-H. Fang and A.E. Yagle, "Two Methods for Toeplitz-plus-Hankel Approximation to a Data Covariance Matrix," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992.

4.7 2-D Spectral Estimation on a Polar Raster

We consider the following spectral estimation problem. A zero-mean homogeneous random field is defined on a polar raster. Given discrete sample values inside a disk of finite radius, estimate the field's power spectral density using a linear prediction model.

Issues arising here include: (1) estimation of covariance lags; (2) extendibility of a finite set of lag estimates into a positive semi-definite covariance extension (required for a meaningful spectral density); and (3) in the lack of performing such an extension, guaranteeing a non-negative spectral density.

Recall that the covariance extension property does not hold on a rectangular raster. However, we give a generalized autocorrelation procedure that guarantees a positive semi-definite covariance extension. It first interpolates the data using Gaussians, computes its Radon transform, and then applies one-dimensional spectral estimation techniques to each slice. We show that if each 1-D set of covariance lags is positive semi-definite, then the extended covariance is also positive semi-definite, so that the 2-D spectral estimate is non-negative and hence meaningful.

The correlation matching property that the extended covariance lags should match the given covariance lags holds in the Radon domain, but not in the spatial domain. We also propose a second algorithm that: (1) matches the given covariance lags; and (2) gives a positive semi-definite extension of them, when this is possible. We also discuss circumstances when this is impossible, shedding some light on 2-D covariance extension.

Details are provided in Appendix G, which consists of the paper: W.-H. Fang and A.E. Yagle, "Two-Dimensional Linear Prediction and Spectral Estimation on a Polar Raster," submitted to *IEEE Trans. Signal Processing*.

4.8 VLSI Implementations of Fast Algorithms

The generalized Levinson and Schur algorithms in Part I are amenable to parallel implementation. The similarity of their recursions to finite difference equations suggests that VLSI implementations for finite differences might be applied to these algorithms. This turns out to be the case, although some changes are required, and certain special cases allow simpler implementations.

Details are provided in Appendix H, which consists of the paper: W.-H. Fang and A.E. Yagle, "A Systolic Architecture for New Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," submitted to *IEEE Trans. Signal Processing*.

4.9 Linear Predictive Coding and Smoothing of Random Fields

The two-dimensional discrete algorithm for random fields on a polar raster has been applied to linear predictive coding of isotropic random fields on a polar raster. One application is in storing images defined on a polar raster (e.g., tomographic data and spotlight synthetic aperture radar data)—storing the residuals, instead of the original image, requires much fewer bits.

The results of using the algorithm are compared with the much simpler procedure of using linear predictive coding independently along each radial slice; this amounts to assuming each radial slice of the image is independent of each other slice. This is of course not true for an isotropic random field, and our results show the significant improvement in image compression ratio using the two-dimensional algorithm.

The two-dimensional algorithm is also applied to smoothing isotropic random fields, in order to reduce noise. This has obvious applications in any setting in which the data consists of noisy observations of a random field. First the prediction filter alone is used to estimate the random field (this is analogous to using previous two-dimensional least-squares filters derived using quarter-plane causality on a rectangular lattice). Then the smoothing filter, derived using the two-dimensional discrete BSK equation, is employed.

The results show considerable improvement (about 8 db) in signal-to-noise ratio when the smoothing filter is used, and about 1 db improvement when the prediction filter is used alone. This demonstrates the importance of the BSK equation—the smoothing filters are indeed necessary.

The results are given in more detail in Appendix I.

5. REFERENCES

1. N. Levinson, "The Wiener RMS Error Criterion in Filter Design and Prediction," *J. Math. Phys.* 25, 261-278 (1947).
2. J. LeRoux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-25(3), 257-259 (1977).
3. S.Y. Kung and Y.H. Hu, "A Highly Concurrent Algorithm and Pipelined Architecture for Solving Toeplitz Systems," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-31(1), 66-75 (1983).
4. P. Delsarte and Y. Genin, "On the Splitting of Classical Algorithms in Linear Prediction Theory," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-35(5), 645-653 (1987).
5. J.H. Justice, "A Levinson-Type Algorithm for Two-Dimensional Wiener Filtering using Bivariate Szego Polynomials," *Proc. IEEE* 65(6), 882-886 (1977).
6. D.G. Manolakis and V.K. Ingle, "Fast Algorithms for 2-D FIR Wiener Filtering and Linear Prediction," *IEEE Trans. Ckts. and Sys.* CAS-34(2), 181-187 (1987).
7. T. Marzetta, "A Linear Prediction Approach to Two-Dimensional Spectral Factorization and Spectral Estimation," Ph.D. Thesis, Dept. of Elec. Eng. and Comp. Sci., Mass. Inst. Tech., Cambridge, Mass., Feb. 1978.
8. J. Rissanen, "Algorithms for Triangular Decomposition of Block Hankel and Toeplitz Matrices with Applications to Factoring Positive Matrix Polynomials," *Math. Comput.* 27, 147-154 (1973).
9. A.E. Yagle, "A Fast Algorithm for Linear Estimation of Three Dimensional Homogeneous Anisotropic Random Fields," 1987 IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, TX, April 6-9, 1987, pp. 1712-1715. Also presented at 1986 IEEE International Symposium on Information Theory, Ann Arbor, MI, October 5-9, 1986.
10. T. Kailath, "Applications of a Resolvent Identity to a Linear Smoothing Problem," *SIAM J. Control* 7, 68-74 (1969).
11. G.A. Merchant and T.W. Parks, "Efficient Solution of a Toeplitz-plus Hankel Coefficient Matrix System of Equations," *IEEE Trans. Acoust., Speech, and Sig. Proc.* ASSP-30(1), 40-44 (1982).

6. PUBLICATIONS SUPPORTED BY THIS GRANT

This project has been quite productive in terms of papers published in top journals. The most prestigious signal processing journal is the *IEEE Transactions on Signal Processing* (formerly *IEEE Transactions on Acoustics, Speech, and Signal Processing*). Four papers are in press, and two more have been submitted to this journal. Another paper has been published in *SIAM Journal of Applied Math.* These papers have been included as Appendices A, B, C, E, F, G, and H. Three other papers have appeared in conference publications; these are included in Appendix J.

6.1 Published Papers Supported by Grant # AFOSR-89-0017

1. A.E. Yagle, "Analogues of Split Levinson, Schur, and Lattice Algorithms for Three-Dimensional Random Field Estimation Problems," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1780-1799, Dec. 1990. Included as Appendix A.
2. A.E. Yagle, "New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-39, no. 11, Nov. 1991. Included as Appendix B.
3. W.-H. Fang and A.E. Yagle, "Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992. Included as Appendix C.
4. W.-H. Fang and A.E. Yagle, "Fast Algorithms for Linear Least-Squares Smoothing Problems in One and Two Dimensions using Generalized Discrete Bellman-Siebert-Krein Resolvent Identities," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992. Included as Appendix E.
5. W.-H. Fang and A.E. Yagle, "Two Methods for Toeplitz-plus-Hankel Approximation to a Data Covariance Matrix," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992. Included as Appendix F.

6.2 Submitted Papers Supported by Grant # AFOSR-89-0017

6. W.-H. Fang and A.E. Yagle, "Two-Dimensional Linear Prediction and Spectral Estimation on a Polar Raster," submitted to *IEEE Trans. Signal Processing*. Included as

Appendix G.

7. W.-H. Fang and A.E. Yagle, "A Systolic Architecture for New Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," submitted to *IEEE Trans. Signal Processing*. Included as Appendix H.

6.3 Papers Presented at Conferences

1. A.E. Yagle, "New Analogues of Split Algorithms for Toeplitz-plus-Hankel Matrices," 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing, Toronto, Canada, May 14-17, 1991, pp. 2253-2256.
2. W.-H. Fang and A.E. Yagle, "Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster," 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, April 3-6, 1990, pp. 2017-2020.
3. A.E. Yagle, "Generalized Levinson and Fast Cholesky Algorithms for Three Dimensional Random Field Estimation Problems," Sixth Multidimensional Signal Processing Workshop, Asilomar, Pacific Grove, CA, September 6-8, 1989. p. 145.

The above three papers are included as Appendix J.

6.4 Other Publications

W.-H. Fang, *Fast Algorithms for Multidimensional Linear Least-Squares Problems*' Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109, June 1991 (210 pages).

6.5 Project Personnel

Principal Investigator: Professor Andrew E. Yagle, Ph.D., M.I.T., 1985. Currently Assistant Professor of Electronic Engineering at The University of Michigan, Ann Arbor.

Research Assistant: Professor Wen-Hsien Fang, Ph.D., University of Michigan, 1991. Currently Assistant Professor of Electronic Engineering at National Taiwan Institute of Technology, Taipei, Taiwan.

APPENDIX A

A.E. Yagle, "Analogues of Split Levinson, Schur, and Lattice Algorithms for Three-Dimensional Random Field Estimation Problems," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1780-1799, Dec. 1990.

Analogue of Split Levinson, Schur, and Lattice Algorithms for Three-Dimensional Random Field Estimation Problems

Andrew E. Yagle

Department of Electrical Engineering and Computer Science

The University of Michigan, Ann Arbor, Michigan 48109-2122

Revised July 1989

ABSTRACT

Fast algorithms for computing the linear least-squares estimate of a three-dimensional random field from noisy observations inside a sphere are proposed. The algorithms can be viewed as three-dimensional analogues of the split Levinson, Schur, and lattice algorithms of linear prediction, since they exploit an (assumed) Toeplitz-plus-Hankel structure of the double Radon transform of the random field covariance. Therefore these algorithms require fewer computations than would solution of the three-dimensional Wiener-Hopf integral equation. Unlike previous generalized Levinson algorithms, no quarter-plane or asymmetric half-plane support assumptions for the filter are necessary; nor is the three-dimensional filtering problem treated as a multichannel (vector) filtering problem.

The algorithms work in three stages. First, the three-dimensional split Schur algorithm computes a potential from the covariance of the random field. This potential is a three-dimensional analogue of the parameter appearing in the split Levinson algorithm. Alternatively, the three-dimensional split lattice algorithm may be used to compute the potential from the canonical spectral factor of the covariance of the observation field. Next, the three-dimensional split Levinson algorithm computes the Radon transform of the three-dimensional prediction filter for estimating the random field on the surface of the sphere of noisy observations. Finally, this filter is used to compute the smoothing filter for estimating the random field inside the sphere of observations. The algorithms generalize known results for isotropic, two-dimensional random fields.

AMS (MOS) subject classification numbers: 93E10, 93E11, 35R30.

Key words: Estimation, fast algorithms, random fields.

This research was supported in part by the Air Force Office of Scientific Research under grant AFOSR-89-0017.

1. Introduction. The problem of computing linear least-squares estimates of three-dimensional random fields from noisy observations is important in such fields as meteorology and processing of time varying images. The enormous amount of computation involved in three-dimensional signal processing requires fast algorithms that exploit any structure in the problem, and that can be parallelized. The obvious choices of fast algorithms for computing estimates from covariance information are three-dimensional generalizations of the one-dimensional Levinson, Schur, and lattice algorithms.

Considerable effort has been applied to generalizing the Levinson algorithm to two dimensions. Although many useful algorithms have been obtained, all of them require some assumptions about the *filter*, i.e., the order in which the data are processed, as opposed to the random field itself. The filters constructed from existing two-dimensional Levinson algorithms are required to have quarter-plane support, or asymmetric half-plane support, or some other such condition, due to the necessity of imposing some well-defined processing order on two-dimensional data. Another approach is to assume line-by-line scanning, so that the two-dimensional estimation problem can be reformulated as a *multichannel* one-dimensional problem, to which the multichannel Levinson algorithm can be applied. Although these assumptions are appropriate for some image processing problems, they are inappropriate for the general estimation problem. Also, extending these conditions to the three-dimensional problem is not trivial.

In this paper we take a different approach. Following [1] and [2] we operate directly on the three-dimensional Wiener-Hopf integral equation, converting it into a three-dimensional differential form. A Radon transform converts this form into a coupled system of partial differential equations that can be propagated, reconstructing the Radon transform of the solution to the integral equation. Alternatively, the differential form can be propagated directly, without resort to the Radon transform. The coupled system of equations can be viewed as a three-dimensional, continuous-parameter analogue of the *split Levinson* algorithm of linear prediction [3].

The potential required to propagate these equations is obtained from a three-dimensional analogue of the split *Schur* algorithm [3]. The split Schur algorithm is initialized using the covariance of the random field. Alternatively, the potential may be computed by initializing a three-dimensional analogue of the split *lattice* algorithm [3]. This algorithm is initialized using the canonical spectral factor of the double Radon transform of the observation field covariance.

All of this is a generalization of what the more familiar one-dimensional Levinson, Schur, and lattice algorithms do, except that the potential function, rather than reflection coefficients, characterizes the optimal filters. Our nomenclature for the three new algorithms thus follows function, rather than form, although there are some marked similarities in form as well.

The procedure proposed here has three stages. The random field covariance is assumed to have a three-dimensional displacement property (equation (3-7) below), so that its double Radon transform has Toeplitz-plus-Hankel structure. Either the random field covariance, or the canonical spectral factor of the covariance of the observation field, may be used to initialize the three-dimensional split Schur or lattice algorithms, respectively. Both of these algorithms compute a three-dimensional version of the potential parameter appearing in the one-dimensional split Levinson, Schur, and lattice algorithms [3]. Next, this potential parameter is used in the three-dimensional split Levinson algorithm to compute the Radon transform of the filter for estimating the random field on the *surface* of a sphere of noisy observations. Finally, the *smoothing* filter for estimating the random field *inside* the sphere of observations is obtained from this filter. A similar approach was used for one-dimensional random fields with Toeplitz covariances in [4], and for two-dimensional isotropic random fields in [5].

It is important to emphasize that NO assumptions are made on the order of processing of the *data*. The *filters* themselves are generated recursively, but the data is not processed in any specific order. The fast algorithm is due entirely to the displacement property

(3-7) of the random field covariance, which is the three-dimensional generalization of the Toeplitz structure exploited by the one-dimensional Levinson and Schur algorithms.

The numerical performance of the new algorithms has not yet been studied, and so they should be viewed as only proposed numerical procedures. However, the insight these algorithms give into the three-dimensional estimation problem, and the way in which they demonstrate how results for one-dimensional and isotropic two-dimensional random fields generalize to three dimensions, is of some interest.

The paper is organized as follow. Section 2 quickly reviews the one-dimensional split Levinson, Schur, and lattice algorithms of [3]. Section 3 specifies the problem in detail, discusses the generalized displacement property (3-7), and quickly reviews the Radon transform. Section 4 derives the differential form of the three-dimensional Wiener-Hopf integral equation, and derives new fast algorithms to obtain the three-dimensional split Levinson, Schur, and lattice algorithms. Section 5 notes how the smoothing filter is obtained, and summarizes the three-stage procedure. Section 6 concludes by summarizing the paper and noting directions for possible future research. Some derivations are relegated to Appendices.

2. The One-Dimensional Split Algorithms. We quickly summarize the one-dimensional split Levinson, Schur, and lattice algorithms of [3], and discuss briefly their scattering interpretations. It should be noted that these algorithms arise in the contexts of inverse scattering [6], network synthesis [7], and orthogonal polynomials [8]. For a historical overview of their place in estimation theory, see [9].

2.1 Classical Levinson Algorithm. Consider the one-dimensional linear prediction problem of estimating the present value of a zero-mean, stationary, discrete-time random process $x(i)$ from observations $\{x(j), i-n \leq j \leq i-1\}$ of its past n values. It is well known that the optimal linear prediction filter coefficients can be obtained using the Levinson algorithm. Let $R(z)$ be the z -transform (where z is the unit delay operator) of one side of

the covariance sequence of $x(i)$. Then the n th-order prediction error filter $A_n(z)$ can be recursively computed as follows [10]:

$$\begin{bmatrix} A_n(z) \\ B_n(z) \end{bmatrix} = \begin{bmatrix} 1 & zk_n \\ k_n & z \end{bmatrix} \begin{bmatrix} A_{n-1}(z) \\ B_{n-1}(z) \end{bmatrix} \quad (2-1a)$$

$$k_{n+1} = -A_n(z)R(z)/(z^{n+1}P_n)|_{z=0} \quad (2-1b)$$

$$P_n = (1 - k_n^2)P_{n-1} \quad (2-1c)$$

$$A_0(z) = B_0(z) = 1 \quad (2-1d)$$

In (2-1b) and the sequel, the notation $f(z)|_{z=0}$ denotes the constant term in the Laurent expansion of $f(z)$. Equations (2-1) also recursively compute the *backwards prediction* error filter $B_n(z)$. This is the error filter for estimating $x(i - n - 1)$ from its future n values $\{x(j), i - n \leq j \leq i - 1\}$.

The $\{k_i, i = 1 \dots n\}$ characterize the optimal prediction filters of all orders up to n : given $\{k_i, i = 1 \dots n\}$, (2-1a) could be used to compute all of the prediction error filters $\{A_i(z), i = 1 \dots n\}$, even though the latter have a total of $n(n + 1)/2$ coefficients. The k_i are called *reflection coefficients*, since equations (2-1) can be implemented on a lattice filter in which signals in one rail are scattered into the other rail, with gain k_i in the i th section of the lattice [11]. This is illustrated in Fig. 1.

Note that the signal propagation in the lattice filter (Fig. 1) is similar to the wave propagation in a one-dimensional scattering medium probed with an impulsive wave at the left end. In this case k_i is the reflection coefficient at the i th interface, which reflects part of the wave travelling in one direction into the wave travelling in the other direction. The connection between one-dimensional scattering and linear prediction has been noted in [6] and [12]; as we shall see, this connection generalizes to three dimensions [13].

In the Levinson algorithm the reflection coefficients k_i are computed using (2-1b), which is called the "inner product" computation. Equations (2-1) require $3n$ multiplications; one-third of these are in (2-1b). Worse, this is a *non-parallelizable* computational

bottleneck; it would be desirable to avoid this computation if possible. This motivates the next two algorithms.

2.2 Classical Schur Algorithm. If the recursions (2-1a) are initialized using $R(z)$ instead of (2-1d), the result is the Schur algorithm [14]:

$$\begin{bmatrix} U_n(z) \\ D_n(z) \end{bmatrix} = \begin{bmatrix} 1 & zk_n \\ k_n & z \end{bmatrix} \begin{bmatrix} U_{n-1}(z) \\ D_{n-1}(z) \end{bmatrix} \quad (2-2a)$$

$$k_{n+1} = -U_n(z)/(zD_n(z))|_{z=0} \quad (2-2b)$$

$$D_0(z) = 1 + R(z); \quad U_0(z) = R(z) \quad (2-2c)$$

The Schur algorithm can be stated in several different forms; we chose this form so that the recursions (2-2a) match (2-1a). In comparing (2-2) with [3], we have $U_k(z) = \sum_{j=k+1}^{n-k} e_{k,j} z^j$ and $D_k(z) = \sum_{j=k}^{n-k} e_{k,k-j} z^j$ for the $e_{i,j}$ and n of [3].

The scattering interpretation of the Schur algorithm is as follows. The Schur algorithm propagates the waves in the lattice structure of Figure 1 resulting from an impulsive initialization (the "1" in (2-2c)) at its left end. Hence it computes the k_i from the *reflection response* $R(z)$.

Note that in the Schur algorithm the k_i are computed using (2-2b), which is not an "inner product" computation (it requires only a single division). Hence the Schur algorithm can be propagated in parallel with the Levinson algorithm, solely for the purpose of computing the reflection coefficients k_i , and thus avoiding the inner product (2-1b) required by the Levinson algorithm alone [15].

2.3 Classical Lattice Algorithm. Now let $X(z)$ be the spectral factor of the two-sided covariance sequence of $x(i)$, i.e.,

$$1 + R(z) + R(1/z) = X(z)X(1/z). \quad (2-3)$$

If the recursions (2-1a) are initialized using $X(z)$ instead of (2-1d), the result is the lattice

algorithm [16]:

$$\begin{bmatrix} F_n(z) \\ G_n(z) \end{bmatrix} = \begin{bmatrix} 1 & zk_n \\ k_n & z \end{bmatrix} \begin{bmatrix} F_{n-1}(z) \\ G_{n-1}(z) \end{bmatrix} \quad (2-4a)$$

$$k_{n+1} = -X_n(z)G_n(1/z)/(z^{n+1}P_n)|_{z=0} \quad (2-4b)$$

$$P_n = (1 - k_n^2)P_{n-1} \quad (2-4c)$$

$$F_0(z) = G_0(z) = X(z) \quad (2-4d)$$

Note that in the lattice algorithm an "inner product" computation (2-4b) is required. Hence its only advantage over the Levinson algorithm is that, given knowledge of $X(z)$ instead of $R(z)$, it avoids the computation (2-3).

The scattering interpretation of the lattice algorithm is as follows [17]. The lattice algorithm propagates the waves in the lattice structure resulting from an impulsive initialization at its *right* end. Hence it computes the k_i from the *transmission* response $X(z)$. The reflection response $R(z)$ and transmission response $X(z)$ are related by (2-3) [18].

2.4 Split Levinson Algorithm. There is some redundancy in the above algorithms. Defining $h_n(z)$ from (2-1a) as

$$h_n(z) = A_n(z) + zB_n(z) \quad (2-5)$$

it may be shown using (2-1) [3] that $h_n(z)$ satisfies the *three-term recurrence*

$$h_{n+1}(z) = (z+1)h_n(z) - za_n h_{n-1}(z) \quad (2-6a)$$

$$h_0(z) = 1+z; \quad h_{-1}(z) = 2 \quad (2-6b)$$

and that a_n may be computed using

$$a_n = v_n^n / v_{n-1}^{n-1}; \quad v_n^n = R(z)h_n(z)/z^{n+1}|_{z=0} \quad (2-7)$$

Equations (2-6) and (2-7) constitute the *split Levinson algorithm*. $h_{-1}(z)$ is defined from (2-5) to initialize the three-term recurrence. The point is that the two coupled recursions

(2-1a) are replaced by the three-term recurrence (2-6). Since (2-6) only requires n multiplications, while (2-1a) requires $2n$, using (2-6) saves 50% of the multiplications. However, note that an "inner product" computation (2-7) is still required at each recursion.

The $\{a_i\}$ characterize the optimal filters of all orders just as the $\{k_i\}$ do; indeed we have [3]

$$a_n = (1 + k_n)(1 - k_{n-1}). \quad (2-8)$$

Also, the quantity $S_n(z) = h_n(z)/w^n$, where $w = z^{1/2}$, satisfies

$$S_{n+1}(z) + S_{n-1}(z) - (w + 1/w)S_n(z) = V_n S_{n-1}(z) \quad (2-9a)$$

$$V_n = 1 - a_n \quad (2-9b)$$

Equation (2-9) has the form of a *discrete Schrodinger equation* [19]. Since a scattering interpretation can be assigned to the lattice-based algorithms, a reformulation of these algorithms in terms of a discrete Schrodinger equation is not surprising.

Note that the scattering potential V_n is simply $1 - a_n$; in the sequel we refer to a_n as a potential. Thus the split algorithms can be interpreted as propagating the field quantities (voltage, pressure, etc.) associated with the scattering medium, while the classical algorithms propagate waves in the scattering medium. For more details see [17].

Since the decomposition of the field quantity into forward and backward travelling waves is not possible in three dimensions, *only* the split algorithms can be generalized to three dimensions. The potential V_n defined in (2-9b) generalizes to three dimensions (see (4-2) below), but there are additional dependencies in it.

2.5 Split Schur Algorithm. Defining $v_n(z)$ from (2-2a) as

$$v_n(z) = U_n(z) + zD_n(z) \quad (2-10)$$

it may be shown using (2-2) [3] that $v_n(z)$ and a_n can be computed using the *split Schur algorithm*:

$$v_{n+1}(z) = (z + 1)v_n(z) - za_nv_{n-1}(z) \quad (2-11a)$$

$$a_n = v_n^n / v_{n-1}^{n-1}; \quad v_n^n = v_n(z) / z^{n+1} |_{z=0} \quad (2-11b)$$

$$v_0(z) = z + R(z) + zR(z); \quad v_{-1}(z) = 1 + 2R(z) \quad (2-11c)$$

$v_{-1}(z)$ is defined from (2-10) to initialize the three-term recurrence. In comparing (2-11) with (18) of [3], note that $v_k(z) = \sum_{j=k+1}^{n-k} v_{k,j-k-1} z^j$ for the $v_{i,j}$ and n of [3].

As with the Levinson algorithm, the split Schur algorithm (2-11) requires only 50% as many multiplications as the classical Schur algorithm (2-2). Also, note that there is no "inner product" computation, so that the split Levinson and Schur algorithms can be propagated together, with the split Schur algorithm replacing the "inner product" (2-7).

2.6 Split Lattice Algorithm. Defining $u_n(z)$ from (2-4a) as

$$u_n(z) = F_n(z) + zG_n(z) \quad (2-12)$$

it may be shown using (2-4) [3] that $u_n(z)$ and a_n can be computed using the *split lattice algorithm*:

$$u_{n+1}(z) = (z+1)u_n(z) - za_n u_{n-1}(z) \quad (2-13a)$$

$$a_n = v_n^n / v_{n-1}^{n-1}; \quad v_n^n = u_n(z)u_n(1/z) |_{z=0} \quad (2-13b)$$

$$u_0(z) = (1+z)X(z); \quad u_{-1}(z) = 2X(z) \quad (2-13c)$$

$u_{-1}(z)$ is defined from (2-12) to initialize the three-term recurrence. Again (2-13a) requires only 50% as many multiplications as (2-4a). However, the "inner product" (2-13b) is still necessary. Given knowledge of $X(z)$, instead of $R(z)$, the split lattice algorithm could be used to compute the k ; without the computation (2-3).

2.7 Continuous Parameter Forms. The continuous-parameter form of the three-term recurrence (2-6) (and also (2-11a) and (2-13a)) is determined by noting that (2-6) is related to a discrete Schrodinger equation (2-9a) by a simple delay. The continuous-parameter Schrodinger equation in the time domain is

$$\left(\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right) h(x, y) = V(x)h(x, y) \quad (2-14)$$

where $h(x, y)$ is the continuous-parameter version of $h_n(z)$ (y is time). Equation (2-14) describes a continuous one-dimensional scattering medium with continuous scattering potential $V(x)$ (the continuous version of (2-9b)). It is also the equation for a vibrating, elastically-based string used in [4] for one-dimensional linear estimation problems.

In the following sections the three-dimensional version of (2-14) is used for three-dimensional linear estimation problems. It should be clear why this can be construed as a three-dimensional, continuous-parameter analogue of the three-term recurrences that constitute the one-dimensional split algorithms.

3. Basic Equations

3.1 Problem Specification. The basic problem is as follows. Let

$$w(x) = s(x) + v(x), \quad x \in R^3 \quad (3-1)$$

be some noisy observations of a zero-mean real-valued random field $s(x)$ having covariance

$$E[s(x)s(y)] = k(x, y). \quad (3-2)$$

$v(x)$ is a zero-mean real-valued white noise field with covariance

$$E[v(x)v(y)] = \delta(x - y) \quad (3-3)$$

and $v(x)$ is uncorrelated with $s(x)$.

We wish to compute the linear least-squares estimate $\hat{s}(x)$ of $s(x)$ given the noisy observations $w(x)$ inside a sphere of radius T , i.e., given $\{w(y), |y| \leq T\}$. To be exact, we wish to compute the conditional mean $E[s(x)|W]$, where W is the Hilbert space spanned by $\{w(y), |y| \leq T\}$. The estimation problem then reduces to computing the optimal filter $g(x, y; T)$, which in turn yields $\hat{s}(x)$ by

$$\hat{s}(x) = \int g(x, y; T)w(y)dy = \int_0^T \int_S g(x, |y|e; T)w(|y|e)|y|^2 de d|y|, \quad |x| \leq T. \quad (3-4)$$

Here S is the unit sphere and $y = |y|e$, where e is a unit vector. de is the differential area on the surface of the unit sphere S ; in standard spherical coordinates $de = \sin \theta d\theta d\phi$. By the orthogonality principle, $g(x, y; T)$ solves the three-dimensional Fredholm integral equation of the second kind

$$k(x, y) = g(x, y; T) + \int_0^T \int_S g(x, re; T) k(re, y) r^2 de dr, \quad 0 \leq |x|, |y| \leq T \quad (3-5)$$

Most of this paper will be concerned with the *intermediate* problem of computing the linear least-squares estimate of $s(x)$ given the noisy observations $\{w(y), |y| \leq |x|\}$. This is the *filtering* problem of estimating $s(x)$ on the *surface* of the sphere of observations. It can also be viewed as the three-dimensional analogue of the linear prediction problem solved by the Levinson algorithm in one dimension. The forward and backward predictors for either end of the segment of observations generalize to the predictors for all points on the surface of the sphere of radius $|x|$.

The optimal filter for this problem is $h(x, y)$, for which the Fredholm integral equation (3-5) becomes the Wiener-Hopf integral equation

$$k(x, y) = h(x, y) + \int_{|z| \leq |x|} h(x, z) k(z, y) dz, \quad |y| \leq |x|. \quad (3-6)$$

Without loss of generality, we define $h(x, y) = 0$ for $|y| > |x|$. $h(x, y)$ can be viewed as the analogue of a continuous-parameter quarter-plane autoregressive filter, except that the causality is defined in terms of $|x|$ and $|y|$, so that there is no "corner" and no ambiguity over in which direction to proceed. In Section 5 we show that $g(x, y; T)$, the ultimate goal, can be obtained easily from $h(x, y)$.

The function $k(x, y)$ is assumed to be positive definite, and it is assumed to have the *generalized displacement property* [13]

$$(\Delta_x - \Delta_y)k(x, y) = 0 \quad (3-7)$$

where Δ_x is the Laplacian with respect to $x \in R^3$, and similarly for Δ_y . Equation (3-7) is a direct generalization of the Toeplitz-plus-Hankel structure exploited by the one-

dimensional Levinson, Schur, and lattice algorithms. The structure (3-7) of the covariance makes possible fast algorithms for solving the integral equation (3-6).

The structure of $k(x, y)$ implied by (3-7) reduces the number of degrees of freedom in the function $k(x, y)$ from six to five. This is still far more general than the case of a *homogeneous* random field having covariance $k(x - y)$ (three degrees of freedom) treated in [1], or the case of an *isotropic* random field having covariance $k(|x - y|)$ (one degree of freedom) treated in [5]. Note that both homogeneous and isotropic random fields are included as special cases of the property (3-7). Note also that not all three components of x and y need refer to spatial variables; a two-dimensional time-varying random field whose spatial covariance satisfies the two-dimensional version of (3-7), and which is also stationary in time, would satisfy (3-7).

3.2 The Radon Transform. The Radon transform will be used extensively throughout this paper. The Radon transform of a function $f(x), x \in R^3$ is defined as

$$\mathcal{R}\{f(\cdot)\}(\tau, e) = \hat{f}(\tau, e) = \int f(x) \delta(\tau - e \cdot x) dx \quad (3-8)$$

so that it is the integral of $f(x)$ over the plane $\tau = e \cdot x$. Note that $\hat{f}(\tau, e) = \hat{f}(-\tau, -e)$. The inverse Radon transform is

$$f(x) = \mathcal{R}^{-1}\{\hat{f}(\cdot, \cdot)\}(x) = -\frac{1}{8\pi^2} \int_S \int_0^\infty \frac{\partial^2}{\partial \tau^2} \hat{f}(\tau, e) \delta(\tau - e \cdot x) d\tau de \quad (3-9)$$

A good treatment of the Radon transform is [20].

An important property of the Radon transform is the *projection-slice theorem* [20]

$$\mathcal{R}\{f(\cdot)\}(\tau, e) = \mathcal{F}_{|k| \rightarrow \tau}^{-1} \{ \mathcal{F}_{x \rightarrow |k|e} \{ f(x) \} \} \quad (3-10)$$

Here $\mathcal{F}_{|k| \rightarrow \tau}^{-1}$ denotes a one-dimensional inverse Fourier transform taking $|k|$ into τ , with $|k|$ extended to negative values by conjugate symmetry. $\mathcal{F}_{x \rightarrow |k|e}$ denotes a three-dimensional Fourier transform taking $x \in R^3$ into $|k|e$.

Another important property, which is the motivation for using the Radon transform in this paper, is [20]

$$\mathcal{R}\{\Delta_x f(\cdot)\}(\tau, e) = \frac{\partial^2}{\partial \tau^2} \mathcal{R}\{f(\cdot)\}(\tau, e). \quad (3-11)$$

Using (3-11), it may be shown that a covariance function satisfying (3-7) will have a Toeplitz-plus-Hankel structure in the double Radon transform domain. To see this, take the double Radon transform of (3-7). This gives

$$\left(\frac{\partial^2}{\partial \tau_1^2} - \frac{\partial^2}{\partial \tau_2^2} \right) \hat{k}(\tau_1, \tau_2, e_1, e_2) = 0 \quad (3-12a)$$

$$\hat{k}(\tau_1, \tau_2, e_1, e_2) = \mathcal{R}_{x \rightarrow \tau_1, e_1} \mathcal{R}_{y \rightarrow \tau_2, e_2} \{k(x, y)\}. \quad (3-12b)$$

where $\mathcal{R}_{x \rightarrow \tau_1, e_1}$ denotes the Radon transform taking $x \in R^3$ into (τ_1, e_1) . This in turn implies the existence of functions $\hat{k}_1(\cdot)$ and $\hat{k}_2(\cdot)$ such that

$$\hat{k}(\tau_1, \tau_2, e_1, e_2) = \hat{k}_1(\tau_1 - \tau_2, e_1, e_2) + \hat{k}_2(\tau_1 + \tau_2, e_1, e_2) \quad (3-13)$$

i.e., that $\hat{k}(\tau_1, \tau_2, e_1, e_2)$ has Toeplitz-plus-Hankel structure. This is the structure that makes possible a fast algorithm solution to (3-6).

4. Three-Dimensional Split Algorithms. In this section fast algorithms for computing the filter $h(x, y)$ from the covariance $k(x, y)$ are derived. These algorithms are three-dimensional analogues of the split algorithms discussed in Section 2. The basic recursion is a three-dimensional generalization of (2-14).

4.1 Differential Form of the Wiener-Hopf Equation

A. The Differential Form in x and y

Applying the operator $(\Delta_x - \Delta_y)$ to the integral equation (3-6) and using the generalized displacement property (3-7), Green's theorem, and the unicity of solution to (3-6) when $k(x, y)$ is positive definite and both $k(x, y)$ and $h(x, y)$ are L^2 yields, after some algebra (see Appendix A),

$$(\Delta_x - \Delta_y)h(x, y) = \int_S V(x, e)h(|x|e, y)|x|^2 de \quad (4-1)$$

where the *non-local filter potential* $V(x, e)$ is defined as

$$V(x, e) = -\frac{2}{|x|^2} \frac{d}{d|x|} |x|^2 h(x, |x|e). \quad (4-2)$$

Note that although the Wiener-Hopf equation (3-6) is only valid for $|y| \leq |x|$, the differential form (4-1) is valid for *all* x and y , since for $|y| > |x|$ we have trivially $0 = 0$. Equation (4-1) is a direct generalization of (2-14) to three dimensions; the only difference is the extra dependence in the potential $V(x, e)$. Even this is not surprising; since $k(x, y)$ has five degrees of freedom, $h(x, y)$ does also, and thus the potential function characterizing the $h(x, y)$ must also have five degrees of freedom.

B. The Three-Dimensional Split Levinson Algorithm of [1]

In [1] the differential form (4-1) was propagated recursively in increasing $|x|$ and $|y| \leq |x|$, yielding $h(x, y)$. At each recursion, the potential $V(x, e)$ was obtained directly from the integral equation (3-6) using $k(x, y)$ and the previously computed values of $h(x, y)$, as follows:

$$V(x, e) = -\frac{2}{|x|^2} \frac{d}{d|x|} |x|^2 \left(k(x, |x|e) - \int_{|z| \leq |x|} h(x, z) k(z, |x|e) dz \right) \quad (4-3)$$

The fast algorithm proposed in [1] for homogeneous random fields is as follows. Equation (4-1) is discretized into a three-term recurrence in increasing $|x|$ and $|y|$, and propagated along with (4-3). The recursion pattern for updating $h(x, y)$ in $|x|$ and $|y|$ using the discretized (4-1) is illustrated in Fig. 2.

Note that (4-3) is necessary to compute $V(x, e)$, since the boundary values $h(x, |x|e)$ and their gradients appearing in (4-2) cannot be computed using (4-1) alone, due to the support of $h(x, y)$. Examination of the recursion pattern illustrated in Fig. 2 makes this clear. This is analogous to the one-dimensional Levinson algorithm, in which k_n is the coefficient of z^n in $A_n(z)$; this coefficient is not computed by (2-1a), so that (2-1b) must also be used.

The computation involved in (4-3), which is a three-dimensional analogue to the "inner product" computation (2-7) in the one-dimensional split Levinson algorithm (but much

worse), is excessive. Furthermore, it would be desirable not to have to compute Laplacians in both x and y . The former computation can be avoided using three-dimensional split Schur or lattice algorithms (see 4.3 and 4.4 below). The transverse part of the Laplacian in y can be eliminated using the Radon transform, as we now demonstrate.

4.2 Three-Dimensional Split Levinson Algorithm

A. The Differential Form in x and t

Since (4-1) holds for all x and y , we can perform a Radon transform of (4-1) taking y into t and e_i . Using (3-11), this yields

$$(\Delta - \frac{\partial^2}{\partial t^2})\hat{h}(x, t, e_i) = \int_S V(x, e)\hat{h}(|x|e, t, e_i)|x|^2 de \quad (4-4)$$

Equation (4-4) describes a continuous three-dimensional scattering medium with *non-local* scattering potential $V(x, e)$. Aside from the non-local nature of the potential, equation (4-4) is a direct generalization of (2-14) to three dimensions.

Next, note that the Laplacian operator Δ can be written as

$$\Delta = \frac{\partial^2}{\partial |x|^2} + \frac{2}{|x|} \frac{\partial}{\partial |x|} + \Delta^T \quad (4-5)$$

where

$$\Delta^T = \frac{1}{|x|^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{|x|^2 \sin^2 \phi} \frac{\partial^2}{\partial \phi^2} \quad (4-6)$$

is the transverse radial Laplacian operator in spherical coordinates. Equation (4-4) can now be written as

$$\left\{ \left(\frac{\partial^2}{\partial |x|^2} + \frac{2}{|x|} \frac{\partial}{\partial |x|} \right) - \frac{\partial^2}{\partial t^2} \right\} \hat{h}(x, t, e_i) = H(x, t, e_i) \quad (4-7)$$

where

$$H(x, t, e_i) = -\Delta^T \hat{h}(x, t, e_i) + \int_S V(x, e)\hat{h}(|x|e, t, e_i)|x|^2 de \quad (4-8)$$

is an auxiliary quantity. Equations (4-7) and (4-8) can be combined into

$$\left(\frac{\partial^2}{\partial |x|^2} - \frac{\partial^2}{\partial t^2} \right) |x| \hat{h}(x, t, e_i) = |x| \left(\int_S V(x, e)\hat{h}(|x|e, t, e_i)|x|^2 de - \Delta^T \hat{h}(x, t, e_i) \right) \quad (4-9)$$

Equation (4-9), which is the heart of the three-dimensional split Levinson and lattice algorithms, should be compared with (2-14).

B. Three-Dimensional Split Levinson Algorithm

The *three-dimensional split Levinson algorithm* consists of (4-9), propagated as a recurrence in discretized $|x|$ and t , and the Radon transform of (4-3). The recursion pattern for updating $\hat{h}(x, t, e_i)$ using the discretized (4-9) is illustrated in Fig. 3. The discretized (4-9) has the same form as the discrete Schrodinger equation (2-9a), except for the following differences:

1. A separate set of recurrences is required for each e_i and $e_x = x/|x|$. The recurrences are independent, and completely parallelizable;
2. The simple multiplication by the potential in (2-9a) and (2-14) becomes a integration over the unit sphere;
3. $\hat{h}(x, t, e_i)$ and $H(x, t, e_i)$ are weighted by $|x|$, since the recursion is in the increasing radius $|x|$ of a sphere;
4. The transverse Laplacian Δ^T must be computed at each recursion. Since this involves only values of $\hat{h}(x, t, e_i)$ on the surface of the sphere of radius $|x|$, this can be done at each recursion. It should be noted that since differentiation is numerically unstable, some regularizing procedure will be needed for this computation.
5. The inverse Radon transform of $\hat{h}(x, t, e_i)$ must be computed at the end of the procedure.

C. Computation of Boundary Values on $t = \pm|x|$

Since $h(x, y) = 0$ for $|y| > |x|$, we have $\hat{h}(x, t, e_i) = 0$ for $t > |x|$. This follows since the plane $t = e_i \cdot y$ passes only through values of y such that $|y| > t > |x|$, and $h(x, y) = 0$ for such values. Since the characteristics of (4-9) are $t = \pm|x|$, the recurrence relation (4-9) will determine $\hat{h}(x, t, e_i)$ for all $-|x| < t < |x|$, and all non-zero values of $\hat{h}(x, t, e_i)$, except for $t = \pm|x|$, will be computed.

The points on the characteristics $t = \pm|x|$ that are not computed in the course of the

recurrences (4-9) can be found using

$$\hat{h}(x, t = -|x|, e_i) = \hat{h}(x, t = |x|, -e_i) \quad (4-10a)$$

$$\left(\frac{\partial}{\partial |x|} + \frac{\partial}{\partial t} \right) |x| \hat{h}(x, t = |x|, e_i) = -|x|^3 V(x, e_i)/2 \quad (4-10b)$$

where the latter is derived in Appendix B using (4-2). Note from (4-10b) that again $V(x, e)$ is not computed as part of the recursions (4-9)—it must be supplied separately, using the Radon transform of (4-3). Also, using (4-9) and (4-10), it can be seen that knowledge of $V(x, e)$ *suffices* to compute $h(x, y)$. Thus the potentials $V(x, e)$ characterize the optimal filters, just as the reflection coefficients do in the one-dimensional case.

As in the one-dimensional case, we now show how three-dimensional split Schur and lattice algorithms may be used to avoid the computation (4-3) in the three-dimensional split Levinson algorithm.

4.3 Three-Dimensional Split Schur Algorithm. The split Schur algorithm must be propagated in x and y , rather than in x and t . The reason for this is that the Schur algorithm propagates the convolution of the prediction error filter and the observation field covariance, which is zero for $|y| < |x|$ by the orthogonality principle. However, the triangularity property of being zero for $|y| < |x|$ does NOT map to the Radon transform domain. This is unlike the Levinson algorithm, in which $h(x, y) = 0$ for $|y| > |x|$ implies $\hat{h}(x, t, e) = 0$ for $t > |x|$. Since the triangularity property is the essential structure of the Schur algorithm (in one or three dimensions), we are forced back to the $x - y$ domain.

A. Differential Form in x and y

In this section we define the residual error filter $\phi(x, y)$, the residual $\chi(x, y)$, and we show that both satisfy the differential form (4-1). In doing so we make use of *propagation of singularities* arguments, in which coefficients of different orders of singularities (delta functions, doublets, etc.) are equated. This can be viewed as equating coefficients of s in Laurent expansions of Laplace transforms; similar reasoning is used to derive transport equations. For more details see [22].

First, we must define the spectral density $M(k, e_1, e_2)$. The structure (3-7) of $k(x, y)$ implies that its double Fourier transform is zero except for its on-shell values. More specifically, the covariance of the observation field $w(x)$ has the property that

$$\mathcal{F}_{x \rightarrow k_1, e_1} \mathcal{F}_{y \rightarrow k_2, e_2} \{\delta(x - y) + k(x, y)\} = M(k_1, e_1, e_2) \delta(|k_1|^2 - |k_2|^2) \quad (4-11)$$

for some function $M(k, e_1, e_2)$.

As an aside, note that the projection-slice theorem (3-10) implies that

$$\begin{aligned} \hat{k}(\tau_1, \tau_2, e_1, e_2) &= \mathcal{F}_{k_1 \rightarrow \tau_1}^{-1} \mathcal{F}_{k_2 \rightarrow \tau_2}^{-1} \{M(k_1, e_1, e_2) \delta(|k_1|^2 - |k_2|^2)\} \\ &= \hat{k}_1(\tau_1 - \tau_2, e_1, e_2) + \hat{k}_2(\tau_1 + \tau_2, e_1, e_2) \end{aligned} \quad (4-12)$$

so that $\hat{k}(\tau_1, \tau_2, e_1, e_2)$ has Toeplitz-plus-Hankel structure in the double Radon transform domain, in agreement with (3-13).

Next, define the *residual filter* $\phi(x, y)$ as

$$\phi(x, y) = \delta(x - y) - h(x, y) \quad (4-13)$$

$\phi(x, y)$ converts the observation field $w(x)$ into the residual field $w(x) - \hat{s}(x|w(y), |y| \leq |x|)$.

Finally, define the *residual* $\chi(x, y)$ as

$$\chi(x, y) = \int \phi(x, z) (\delta(z - y) + k(z, y)) dz \quad (4-14)$$

$\chi(x, y)$ is the convolution of the prediction error filter and the observation field covariance, just as in the one-dimensional case. Using Parseval's theorem on (4-14), we have

$$\begin{aligned} \hat{\chi}(x, k_2, e_2) &= \mathcal{F}_{y \rightarrow k_2, e_2} \{\chi(x, y)\} = \int \int_S \phi(x, k_3, e_3) M^*(k_2, e_2, e_3) \delta(|k_2|^2 - |k_3|^2) |k_3|^2 de_3 dk_3 \\ &= \int_S \phi(x, k_2, e_3) M^*(k_2, e_2, e_3) |k_2|^2 de_3 \end{aligned} \quad (4-15)$$

We now show that $\phi(x, y)$ and $\chi(x, y)$ both satisfy (4-1). Apply the operator $\Delta_x - \Delta_y$ to (4-13), and recall that $h(x, y) = h(x, y)1(|x| - |y|)$, where $1(\cdot)$ is the unit step or

Heaviside function (this expresses the support constraint for $h(x, y)$). Equating coefficients of singularities (delta functions) and using (4-1) and (4-2) shows that $\phi(x, y)$ satisfies (4-1). Fourier transforming (4-1) with respect to y , and a linearity argument using (4-15), show that $\chi(x, y)$ satisfies (4-1) as well.

B. Recursions

By the orthogonality principle we have $\chi(x, y) = 0$ for $|y| < |x|$. Then, since $\phi(x, y)$ contains an impulse $\delta(x - y)$, $\chi(x, y)$ must contain one also, and thus it has the form

$$\chi(x, y) = \delta(x - y) + v(x, y)1(|y| - |x|) \quad (4-16)$$

Inserting (4-16) into the differential form (4-1) and equating coefficients of singularities results in

$$(\Delta_x - \Delta_y)v(x, y) = \int_S V(x, e)v(|x|e, y)|x|^2 de \quad (4-17a)$$

$$V(x, e) = -2 \left(\frac{\partial}{\partial |x|} + \frac{\partial}{\partial |y|} \right) v(x, y = |x|e) \quad (4-17b)$$

Equations (4-17) constitute the recursions for the *three-dimensional split Schur algorithm*. $v(x, y)$ is propagated in increasing $|x|$ and $|y| > |x|$ using (4-17a), and $V(x, e)$ reconstructed using (4-17b). The recursion pattern for updating $v(x, y)$ is illustrated in Fig. 4. Note that $V(x, e)$ is computed directly by the recursions (4-17a); no "inner product" computation is required. The computed $V(x, e)$ is then inserted in (4-1) or (4-9) to compute $h(x, y)$ via the three-dimensional split Levinson algorithm, avoiding the "inner product" (4-3).

C. Initialization

The split Schur algorithm is initialized by setting $|x| = 0$ in (4-14). This results in

$$v(0, y) = k(0, y) \quad (4-18a)$$

$$\left(\frac{\partial}{\partial |x|} + \frac{\partial}{\partial |y|} \right) v(0, y) = \left(\frac{\partial}{\partial |x|} + \frac{\partial}{\partial |y|} \right) k(0, y) \quad (4-18b)$$

Note that the dependence of $k(x, y)$ on $e = x/|x|$ for small $|x|$ is needed in (4-18b). This ensures the five degrees of freedom in the data necessary to compute $V(x, e)$, which also has five degrees of freedom.

D. Interpretation

The split Schur algorithm propagates $\chi(x, y)$, which is zero for $|y| < |x|$ by the orthogonality principle. This is the stochastic interpretation. However, the scattering interpretation is more illuminating. The form of (4-16) suggests that $\chi(x, y)$ results from initializing (4-1) with an impulse in x and y at the origin $x = 0$, which spreads out in increasing $|x|$ along the characteristic $|y| = |x|$ (note that $|y|$ plays the role of time). The jump in $v(x, y)$, the non-impulsive part of $\chi(x, y)$, on this characteristic yields information about the scattering potential $V(x, e)$.

All of this is analogous to the one-dimensional Schur algorithm; note that for this type of scattering experiment, the non-local nature of $V(x, e)$ does not affect the support of $\chi(x, y)$. Note also that since both the excitation and the measurement takes place at the origin, this is a reflection-type inverse scattering problem, as opposed to the transmission-type problem solved by the lattice algorithm below.

This algorithm is called a three-dimensional Schur algorithm for the following reasons:

1. It solve a reflection-type inverse scattering problem;
2. It propagates the residuals $v(x, y)$, whose triangular structure stems from the orthogonality principle;
3. It is initialized directly with the random field covariance $k(x, y)$;
4. It performs a spectral factorization (see (4-28) below).

4.4 Three-Dimensional Split Lattice Algorithm. In this section we derive *two forms* of the three-dimensional split lattice algorithm. One form is propagated in increasing $|x|$, is initialized directly using the spectral factor of the covariance $k(x, y)$, and requires an "inner product." The other form is propagated in *decreasing* $|x|$, is initialized at large

$|x|$ using the spectral factor, and *does not require* an "inner product." This second form, which has no one-dimensional counterpart, exists because the potential $V(x, e)$ in (4-2) is non-local.

A. Spectral Factorization

Since $\delta(x-y) + k(x, y)$ is the covariance of $w(x)$, and we are now interested in deriving a lattice algorithm, we consider the spectral factorization of the spectral density $M(k, e_1, e_2)$ defined in (4-11) into (compare to (2-3))

$$M(k, e_1, e_2) = \int_S F(k, e_1, e_3) F(k, e_3, e_2)^* de_3 \quad (4-19)$$

where $F(k, e_1, e_2)$ is analytic in k in the lower half-plane. This factorization is a Riemann-Hilbert problem (see (4.1) and (6.9) of [21]); Section 2 of [21] proves, subject to assumptions about $k(x, y)$ already made, that this problem has a unique solution.

In practice, the spectral factorization (4-19) would never be performed; unless $F(k, e_1, e_2)$ is known initially in lieu of $k(x, y)$, there is no point in using the split lattice algorithm. In this case the split Schur algorithm, initialized using $k(x, y)$, is to be preferred.

B. Recursions

Let $\phi(x, k, e_1) = \mathcal{F}_{y \rightarrow k, e_1} \{ \phi(x, y) \}$, where $\phi(x, y)$ is the residual filter defined in (4-13). Define $\psi(x, k, e)$ using

$$\psi(x, k, e_1) = \int_S F(k, e_1, e_2) \phi(x, k, e_2) de_2 \quad (4-20a)$$

$$\phi(x, k, e_1) = \int_S F^{-1}(k, e_1, e_2) \psi(x, k, e_2) de_2 \quad (4-20b)$$

where $F^{-1}(k, e_1, e_2)$ is the inverse kernel to $F(k, e_1, e_2)$.

Let $\hat{\phi}(x, t, e_i)$ and $\hat{\psi}(x, t, e_i)$ be the inverse Fourier transforms of $\phi(x, k, e_i)$ and $\psi(x, k, e_i)$. We showed previously that $\phi(x, y)$ satisfies (4-1), hence $\hat{\phi}(x, t, e_i)$ satisfies (4-9). And (4-20a) and a linearity argument shows that $\hat{\psi}(x, t, e_i)$ also satisfies (4-9).

Since $F(k, e_1, e_2)$ is the canonical spectral factor of $\delta(x-y) + k(x, y)$, the form of (4-9) (specifically its characteristic at $t = -|x|$) implies that $\hat{\psi}(x, t, e_i)$ has the form

$$\hat{\psi}(x, t, e_i) = \delta(t - e_i \cdot x) + u(x, t, e_i) 1(t + |x|) \quad (4-21)$$

To see this, note that $\hat{\phi}(x, t, e_i)$ has support in t on $[-|x|, |x|]$ (since $\hat{h}(x, t, e)$ does), and $F(k, e_1, e_2)$ is causal in the t domain. Examination of the convolution in t implied by (4-20a) shows that $\hat{\psi}(x, t, e_i)$ has support in t on $[-|x|, \infty]$, yielding (4-21).

Inserting (4-21) into (4-9) and once again equating coefficients of singularities results in

$$\left(\frac{\partial^2}{\partial |x|^2} - \frac{\partial^2}{\partial t^2} \right) |x| u(x, t, e_i) = |x| \left(\int_S V(x, e) u(|x|e, t, e_i) |x|^2 de - \Delta^T u(x, t, e_i) \right) \quad (4-22a)$$

$$V(x, -e_i) = \frac{2}{|x|^3} \frac{d}{dx} |x| u(x, t = -|x|, e_i) = \frac{2}{|x|^3} \left(\frac{\partial}{\partial x} - \frac{\partial}{\partial t} \right) |x| u(x, t = -|x|, e_i) \quad (4-22b)$$

Equation (4-22a) has the same form as the three-dimensional split Levinson algorithm (4-9), and it may be propagated in discretized $|x|$ and t in the same way that (4-9) was. Equations (4-22) appear in [23] as a proposed fast algorithm for solving inverse scattering problems with non-local potentials. Compare (4-22b) with (4-2) and (4-10b).

C. Two Three-Dimensional Lattice Algorithms

Equations (4-22) constitute the recursions for the *three-dimensional split lattice algorithm*. By propagating them in either increasing or decreasing $|x|$, we get two different three-dimensional lattice algorithms. The recursion patterns for updating $u(x, t, e_i)$ using (4-22) are illustrated in Fig. 5.

The first algorithm proceeds by initializing $u(x, t, e_i)$ at the origin $x = 0$, using (4-23) below, and propagating (4-22) in *increasing* $|x|$ and $t > -|x|$. Fig. 5 shows that $V(x, e)$ is not computed directly by (4-22); an "inner product" combining (4-20b) with the Radon transform of (4-3) is needed. Because of this, the first algorithm is computationally inferior to the second algorithm. However, it is analogous to the one-dimensional split lattice algorithm.

The second algorithm proceeds by initializing $u(x, t, e_i)$ for *large* $|x|$, using (4-26) below, and propagating (4-22) in *decreasing* $|x|$ and $t > -|x|$. The advantage of this form is that $V(x, e)$ is now computed directly using (4-22) (see Fig. 5); no "inner product" is needed. This makes it clearly superior for computation.

D. Initialization

The first form of the algorithm is initialized by setting $x = 0$ in (4-20a) and using (4-21):

$$u(0, t, e_1) = \mathcal{F}^{-1}\{\psi(0, k, e_1) - 1\} = \mathcal{F}^{-1}\left\{\int_S F(k, e_1, e_2) de_2 - 1\right\} \quad (4-23)$$

Note that (4-23) can be viewed as *transmission* scattering data at the origin.

To initialize the second form of the algorithm we use scattering arguments, following [13] and [24]. Equation (4-21) shows that $\hat{\psi}(x, t, e_i)$ consists of a probing impulsive plane wave $\delta(t - e_i \cdot x)$ and a resulting scattered field $u(x, t, e_i)$. For $|x| > T$, we can write this in the frequency domain as

$$\psi(x, k, e_2) = \int_S S(k, e_1, e_2) \psi(x, -k, -e_1) de_1 \quad (4-24)$$

where $S(k, e_1, e_2)$ is a scattering operator. For large $|x|$, $\psi(x, -k, -e_1)$ represents solely the probing plane wave by time causality. Equations (4-20) and (4-24) combine to give [13],[24]

$$S(k, e_1, e_2) = \int_S F^{-1}(k, e_1, e_3) F(-k, e_3, e_2) de_3 \quad (4-25)$$

Inserting (4-25) in (4-24) allows the second algorithm to be initialized *for large* $|x|$ using

$$\psi(x, k, e_2) = \int_S \int_S F^{-1}(k, e_1, e_3) F(-k, e_3, e_2) e^{-ik e_1 \cdot x} de_3 de_1, \quad |x| \rightarrow \infty. \quad (4-26)$$

E. Stochastic Interpretation

The various quantities appearing in the above derivations all have important stochastic interpretations. We briefly summarize them here; for more details see [13]. $\phi(x, y)$ is the *residual filter* that converts the observation field $w(x)$ into the residual field $r(x) = w(x) - \hat{s}(x|w(y), |y| \leq |x|)$. This residual field can be decorrelated on the circle $|y| = |x|$ to give an innovations field. $\chi(x, y)$ is the *residual*, or the difference between the left and right sides of the Wiener-Hopf equation (3-6); for $|y| < |x|$ $\chi(x, y) = 0$ by the orthogonality principle.

$F^{-1}(k, e_1, e_2)$ is the transfer function of a whitening filter that whitens $w(x)$ to a white field $\nu(x)$, while $F(k, e_1, e_2)$ is the transfer function of a modelling filter that transforms the white field $\nu(x)$ back into $w(x)$. Note that $F(k, e_1, e_2)$ and $F^{-1}(k, e_1, e_2)$ are both causal in the double Radon transform domain: $\hat{F}(\tau, e_1, e_2) = \mathcal{F}^{-1}\{F(k, e_1, e_2)\}$ is causal in τ . However, they are NOT triangular in the spatial domain: $\tilde{F}(x, y) = \mathcal{F}^{-1}\mathcal{F}^{-1}\{F(k, e_1, e_2)\} \neq 0$ for $|y| > |x|$. Hence the white field $\nu(x)$ is not an innovations field; it cannot be obtained from causal filtering of $\{w(x)\}$.

$\psi(x, k, e_i)$ filters $\nu(x)$ into $r(x)$, as shown by (4-20a): First $F(k, e_1, e_2)$ dewhitens $\nu(x)$ to $w(x)$, then $\phi(x, y)$ rewhitens $w(x)$ to $r(x)$. Also, note that (4-26) is the generalization of a well-known one-dimensional result [24].

We now show that the three-dimensional split Schur algorithm performs the spectral factorization (4-19). For large $|x|$, (4-20b) becomes

$$\phi(x, k, e_2) \simeq \int_S F^{-1}(k, e_1, e_2) e^{-ik_{e_1} \cdot x} de_1 \quad (4-27)$$

Inserting (4-27) in (4-15) yields

$$\chi(x, k, e_2) \simeq \int_S F^*(k, e_1, e_2) e^{-ik_{e_1} \cdot x} de_1 \quad (4-28)$$

so that $\chi(x, k, e_2)$, propagated by the Schur algorithm, converges to the spectral factor of the observation field. This is a direct generalization of one-dimensional results [14].

These algorithms are called three-dimensional lattice algorithms for the following reasons:

1. It solve a transmission-type inverse scattering problem;
2. It is initialized directly using the spectral factor $F(k, e_1, e_2)$ of the random field covariance $k(x, y)$;
3. $u(x, t, e_i)$ has the same support $t > -|x|$ as the one-dimensional lattice algorithm.

5. Computation of the Smoothing Filter $g(x, y; T)$

5.1 Computation of $g(x, y)$ from $h(x, y)$. We now specify the third and final stage of the estimation problem: the determination of the smoothing filter $g(x, y; T)$ from $h(x, y)$. Recall that $g(x, y; T)$ is the filter for estimating the random field $s(x)$ from the set of observations $\{w(y), |y| < T\}$. Therefore the computation of $g(x, y; T)$ completes the solution to the original estimation problem. The material of this section is taken from [1], and generalizes results in [4] and [5].

Recall that $g(x, y; T)$ satisfies the Fredholm integral equation (3-5), while $h(x, y)$ satisfies the Wiener-Hopf integral equation (3-6). Taking the partial derivative with respect to T of (3-5), and again using the linearity and unicity of solution properties of (3-6) (the argument is similar to that in Appendix A) results in the differential form

$$\frac{\partial}{\partial T} g(x, y; T) = - \int_S g(x, Te; T) h(Te, y) T^2 de. \quad (5-1)$$

Equation (5-1) allows $g(x, y; T)$ to be computed recursively from $h(x, y)$. The boundary value $g(x, Te; T)$, the only missing value when (5-1) is propagated recursively in increasing T for all $0 < |x|, |y| < T$, can be computed separately by setting $y = Te$ in (3-5). This yields

$$g(x, Te; T) = k(x, Te) - \int_0^T \int_S g(x, re_r; T) k(re_r, Te) r^2 de_r dr \quad (5-2)$$

which computes $g(x, Te; T)$ from already-computed $g(x, z; T)$, $|z| < T$ and known $k(x, y)$.

5.2 Summary of Entire Procedure. The complete, procedure for computing $g(x, y)$ from $k(x, y)$ or $F(k, e_1, e_2)$ is as follows:

1. If $k(x, y)$ is known, use it in (4-18) to initialize the split Schur algorithm. If the spectral factor $F(k, e_1, e_2)$ is known, use it in (4-26) to initialize the split lattice algorithm;
2. Propagate the split lattice algorithm in *decreasing* $|x|$, computing $V(x, e)$ as the recursion proceeds. Alternatively, propagate the split Schur algorithm in *increasing* $|x|$;
3. Propagate the split Levinson algorithm in *increasing* $|x|$, using the potential $V(x, e)$ computed in step 2;

4. Compute $h(x, y) = \mathcal{R}^{-1}\{\hat{h}(x, t, e_i)\}$. This corresponds to the prediction filter in the one-dimensional Levinson algorithm, with the prediction order being the size T of the sphere of observations;
5. Compute $g(x, y)$ from $h(x, y)$ by propagating (5-1) and (5-2).

6. Conclusion. Three-dimensional split Levinson, Schur, and lattice algorithms for the three-dimensional random field least-squares estimation problem have been obtained. These algorithms directly solve the three-dimensional Wiener-Hopf integral equation satisfied by the optimal filter, and make no assumptions about the order in which the three-dimensional data are to be processed. The algorithms are fast since they exploit the Toeplitz-plus-Hankel structure of the double Radon transform of the covariance of the observation field $w(x)$, to reduce the amount of computation necessary to solve the integral equation.

The one-dimensional split algorithms are three-term recurrences that are equivalent (within a delay) to a discretization of a one-dimensional Schrodinger equation in the time domain. The three-dimensional algorithms of Section 4 are equivalent to three-dimensional Schrodinger equations in the time domain, which is why these algorithms are referred to as three-dimensional split algorithms. The connections between three-dimensional estimation and inverse scattering problems has been detailed elsewhere [13]; it is worth noting here that the Wiener-Hopf integral equation (3-6) and the differential form (4-1) both appeared in an inverse scattering context in [24] and [25].

Some issues that need further research are as follows. The non-local potential $V(x, e)$ complicates matters enormously, since it has no one-dimensional analogue and introduces non-causality. It would be very desirable to be able to characterize the set of covariance functions $k(x, y)$, or spectral factors $F(k, e_1, e_2)$, associated with a *local* potential $V(x, e) = V(x)\delta(x/|x| - e)$. This would lead to causal algorithms much more like the one-dimensional algorithms. Elements of this set would have three degrees of freedom, like the set of

covariance functions associated with homogeneous random fields. We note here that this is a major unsolved problem in inverse scattering theory; an estimation viewpoint may well be more appropriate for solving this problem. Another issue is the numerical performance of these algorithms.

Acknowledgment. The author wishes to extend his deepest thanks to the referees who critiqued the first version of this paper in exhaustive detail. Their time, effort and interest are greatly appreciated. This research was supported in part by the Air Force Office of Scientific Research under grant AFOSR-89-0017.

Appendix A

Derivation of the Differential Form (4-1)

Applying the operator $\Delta_x - \Delta_y$ to both sides of (3-6) and using the three-dimensional displacement property (3-7) results in

$$(\Delta_x - \Delta_y)h(x, y) + \Delta_x \int_{|z| \leq |x|} h(x, z)k(z, y)dz - \int_{|z| \leq |x|} h(x, z)\Delta_x k(z, y)dz = 0 \quad (A1)$$

where (3-7) has been used again in the last term. Simplifying the middle term and using Green's identity on the last term gives

$$(\Delta_x - \Delta_y)h(x, y) + \int_{|z| \leq |x|} \{(\Delta_x - \Delta_z)h(x, z)\}k(z, y)dz = \int_S V(x, e)k(|x|e, y)|x|^2 de \quad (A2)$$

where $V(x, e)$ is defined by (4-2).

In the integral equation (3-6) let $x = |x|e$, multiply by $|x|^2 V(x, e)$, and integrate over S . This gives

$$\int_S V(x, e)h(|x|e, y)|x|^2 de + \int_{|z| \leq |x|} \int_S V(x, e)h(|x|e, z)k(z, y)|x|^2 de dz = \int_S V(x, e)k(|x|e, y)|x|^2 de. \quad (A3)$$

Comparing (A2) and (A3) shows that these integral equations have the same form, and are therefore the same equation. Since the operator

$$K : a(t) \rightarrow b(t) = \int_{|s| \leq |t|} k(t, s)a(s)ds \quad (A4)$$

defined by the covariance kernel $k(x, y)$ is self-adjoint and non-negative definite, the operator $K + I$ is invertible. This means that the solution of the integral equation (3-6) exists and is unique. By linearity, therefore, the solutions of the integral equations (A2) and (A3) must be identical. Equation (4-1) follows.

Appendix B

Derivation of Equation (4-10b)

Rewrite (4-2) as

$$V(x, y)\delta(|x| - |y|) = -2 \left(\frac{\partial}{\partial x} + \frac{1}{|x|} + \frac{1}{|y|} + \frac{\partial}{\partial y} \right) h(x, y)\delta(|x| - |y|). \quad (B1)$$

Using the property of the Radon transform that

$$\mathcal{R} \left\{ \left(\frac{\partial}{\partial x} + \frac{1}{|x|} \right) f(x) \right\} = \tau \frac{\partial}{\partial \tau} \mathcal{R} \left\{ \frac{f(x)}{|x|} \right\} \quad (B2)$$

a Radon transform of (B1) taking y into t and e_i yields

$$\int V(x, y)\delta(|x| - |y|)\delta(t - e_i \cdot y)dy = -2 \left(\frac{\partial}{\partial x} + \frac{1}{|x|} + \frac{t}{|x|} \frac{\partial}{\partial t} \right) \hat{h}(x, t, e_i)\delta(t - |x|). \quad (B3)$$

where the $\delta(|x| - |y|)$ has been used to convert $1/|y|$ to $1/|x|$ and then pull it outside of the Radon transform with respect to y . Setting $t = |x|$ reduces the left side of (B3) to $|x|^2 V(x, e_i)$, and quickly yields equation (4-10b).

References

1. A.E. Yagle, "A Fast Algorithm for Linear Estimation of Three-Dimensional Homogeneous Anisotropic Random Fields," *Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc.*, Dallas, TX, April 1987, pp. 1712-1715.
2. A.E. Yagle, "Generalized Levinson and Fast Cholesky Algorithms for Three-Dimensional Random Field Estimation Problems," *Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc.*, New York, April 1988, pp. 1060-1063.

3. P. Delsarte and Y. Genin, "On the Splitting of Classical Algorithms in Linear Prediction Theory," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-35(5), 645-653 (1987).
4. B.C. Levy and J.N. Tsitsiklis, "Linear Estimation of Stationary Stochastic Processes, Vibrating Strings, and Inverse Scattering," Tech. Report #LIDS-P-1155, Laboratory for Information and Decision Systems, M.I.T., 1982.
5. B.C. Levy and J.N. Tsitsiklis, "A Fast Algorithm for Linear Estimation of Two-Dimensional Isotropic Random Fields," *IEEE Trans. Info. Th.* IT-31(5), 635-644 (1985).
6. A.M. Bruckstein and T. Kailath, "Inverse Scattering for Discrete Transmission Line Models," *SIAM Review* 29(3), 359-389 (1987).
7. P. Dewilde, A. Vieira and T. Kailath, "On a Generalized Szego-Levinson Realization Algorithm for Optimal Linear Predictors Based on a Network Synthesis Approach," *IEEE Trans. Ckts. and Sys.* CAS-25, 663-675 (1978).
8. T. Kailath, A. Vieira, and M. Morf, "Inverses of Toeplitz Operators, Innovations, and Orthogonal Polynomials," *SIAM Review* 20(1) 106-119 (1978).
9. T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Info. Th.* IT-20, 145-181 (1974).
10. N. Levinson, "The Wiener RMS error criterion in filter design and prediction," *J. Math. Phys.* 25, 261-278 (1947).
11. B. Friedlander, "Lattice Filters for Adaptive Processing," *Proc. IEEE* 70(8), 829-867 (1982).
12. P. Dewilde, J.T. Fokkema, and I. Widya, "Inverse Scattering and Linear Prediction, the Time Continuous Case," in *Stochastic Systems: The Mathematics of Filtering and Identification and Applications*, M. Hazewinkel and J.C. Willems eds., D. Reidel, Dordrecht, Holland, 1981, pp. 351-382.
13. A.E. Yagle, "Connections between Three-Dimensional Inverse Scattering and the Linear Least-Squares Estimation of Random Fields," *Acta Appl. Mathe.* 13, 267-289

(1988).

14. J. LeRoux and C. Gueguen, "A Fixed Point Computation of Partial Correlation Coefficients," *IEEE Trans. ASSP* ASSP-25, 257-259 (1977).
15. S.Y. Kung and Y.H. Hu, "A Highly Concurrent Algorithm and Pipelined Architecture for Solving Toeplitz Systems," *IEEE Trans. Acoust., Speech, Sig. Proc.* ASSP-31(1), 66-75 (1983).
16. G. Cybenko, "A General Orthogonalization Technique with Applications to Time Series Analysis," *Math. of Comput.* 40, 323-336 (1983).
17. A.E. Yagle, "Fast Algorithms for Estimation and Signal Processing: An Inverse Scattering Framework," *IEEE Trans. Acoust., Speech, and Sig. Proc.* ASSP-37(6), 957-959 (1989).
18. G. Kunetz and I. d'Erceville, "Sur Certaines Propriétés d'une Onde Acoustique Plane de Compression dans un Milieu Stratifié," *Annales de Geophysique* 18, 351-359 (1962).
19. A.M. Bruckstein and T. Kailath, "On Discrete Schrodinger Equations and their Two-Component Wave Equation Equivalents," *J. Math. Phys.* 28(12), 2914-2924 (1987).
20. S.R. Deans, *The Radon Transform and Some of its Applications*, Wiley, New York, 1983.
21. R.G. Newton, "Inverse Scattering. IV. Three Dimensions: Generalized Marchenko Construction with Bound States, and Generalized Gel'fand-Levitan Equations," *J. Math. Phys.* 23(4), 594-604 (1982).
22. R. Burridge, "The Gelfand-Levitan, the Marchenko, and the Gopinath-Sondhi Integral Equations of Inverse Scattering Theory, Regarded in the Context of Inverse Impulse-Response Problems," *Wave Motion* 2, 305-323 (1980).
23. A.E. Yagle, "Differential and Integral Methods for Three-Dimensional Inverse Scattering Problems with a Non-Local Potential," *Inverse Problems* 4(2), 549-566 (1988).
24. R.G. Newton, "Inverse Scattering. II. Three Dimensions," *J. Math. Phys.* 21(7), 1698-1715 (1980).

25. I. Kay and H.E. Moses, "A Simple Verification of the Gelfand-Levitan Equation for the Three-Dimensional Scattering Problem," *Comm. Pure Appl. Math.* 14, 435-445 (1961).
26. T. Kailath, *Modern Signal Processing*, Hemisphere, New York, 1985, p. 85.

Figure Headings

FIG. 1. Lattice filter implementing (2-1) [26].

FIG. 2. Recursion pattern for updating $h(x, y)$ in the three-dimensional Levinson algorithm of [1].

FIG. 3. Recursion pattern for updating $h(x, t, e_i)$ in the three-dimensional split Levinson algorithm.

FIG. 4. Recursion pattern for updating $v(x, y)$ in the three-dimensional split Schur algorithm.

FIG. 5. Recursion pattern for updating $u(x, t, e_i)$ in the three-dimensional split lattice algorithm.

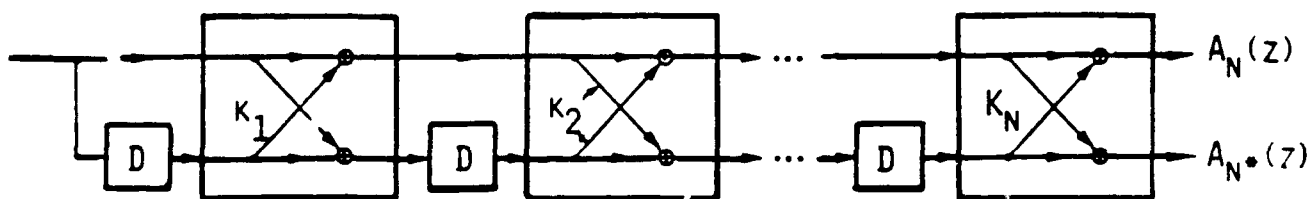
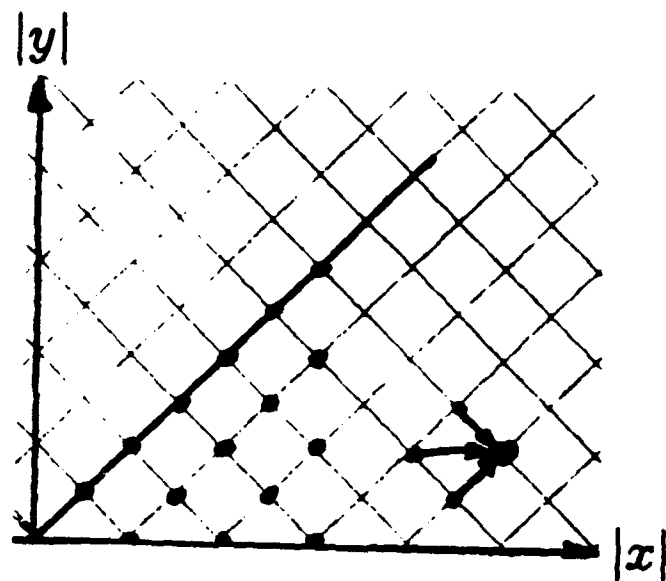


FIG. 1

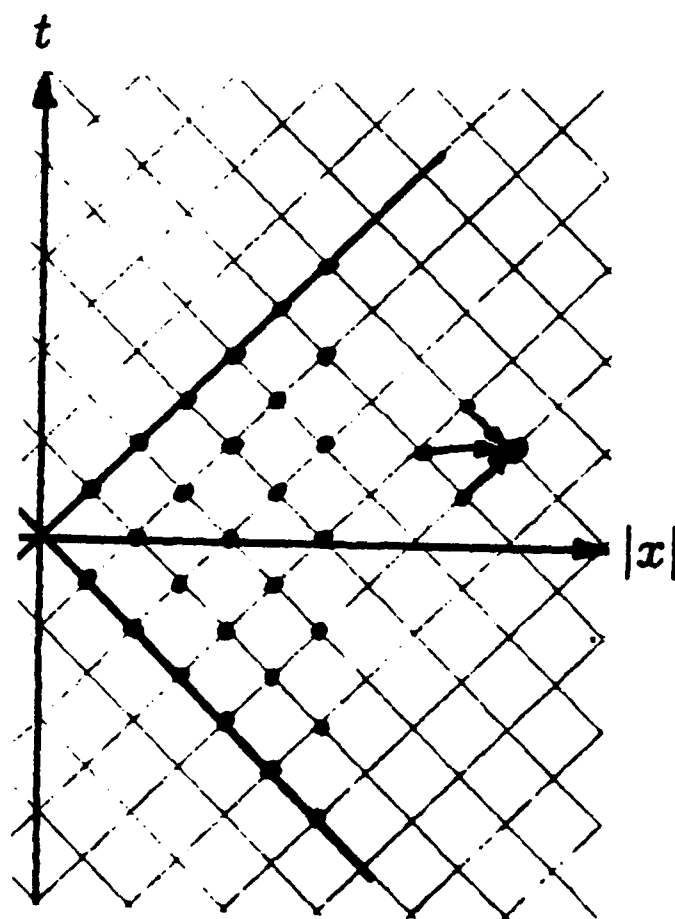
SPLIT LEVINSON ALGORITHM



$$h(x, y)$$

FIG. 2

SPLIT LEVINSON ALGORITHM



$$\hat{h}(x, t, e_i)$$

FIG. 3

SPLIT SCHUR ALGORITHM

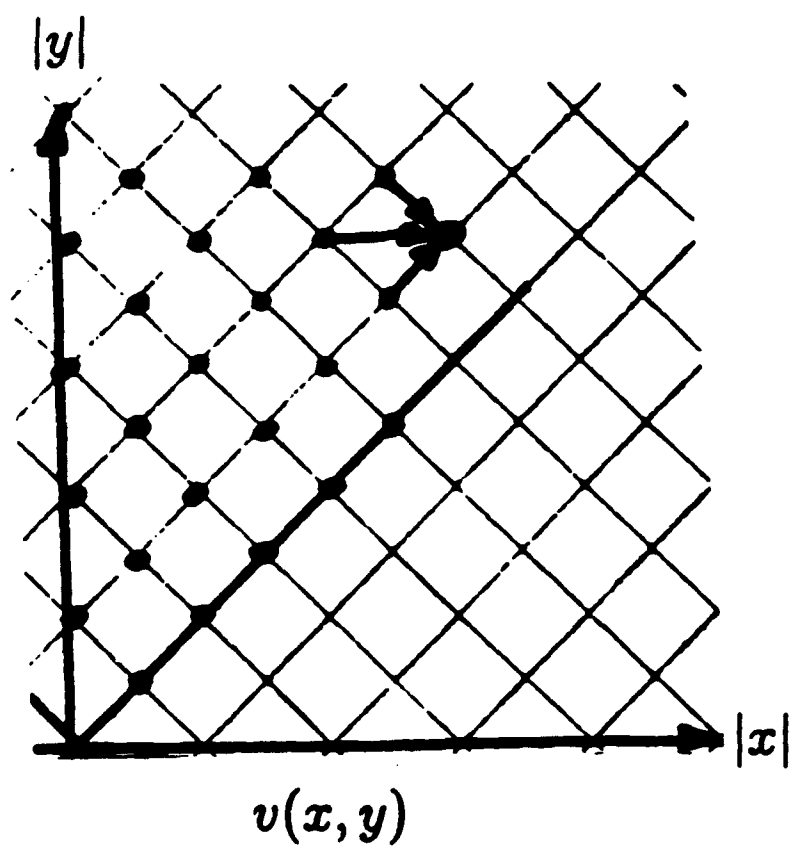


FIG. 4

FIG. 5

APPENDIX B

A.E. Yagle, "New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-39, no. 11, Nov. 1991.

New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices

Andrew E. Yagle

Department of Electrical Engineering and Computer Science

The University of Michigan

Ann Arbor, Michigan 48109-2122

Revised August 1989

ABSTRACT

New fast algorithms for solving arbitrary Toeplitz-plus-Hankel systems of equations are presented. The algorithms are analogues of the split Levinson and Schur algorithms, although the more general Toeplitz-plus-Hankel structure requires that the algorithms be based on a four-term recurrence; relations with previous split algorithms are noted. The algorithms require roughly half as many multiplications as previous fast algorithms for Toeplitz-plus-Hankel systems.

EDICS number: 5.2.4. Phone number: (313) 763-9810

I. INTRODUCTION

Toeplitz-plus-Hankel systems of equations have many important applications. The linear prediction problem for nonstationary random processes with Toeplitz-plus-Hankel covariance functions is one; the recently-developed two-sided autoregressive spectral estimation procedure [1] is another. Toeplitz-plus-Hankel systems also appear in linear-phase prediction filter design [2], the Hildebrand-Prony spectral line estimation procedure [3], and PADE approximation to the cosine series expansion of an even function [4]. The continuous-time counterpart (an integral equation with a Toeplitz-plus-Hankel kernel) appears in atmospheric scattering [5] and rarefied gas dynamics [6].

Fast algorithms for solving Toeplitz-plus-Hankel systems have appeared in [7], in which the Toeplitz-plus-Hankel system is reformulated as a block-Toeplitz system, and [8], in which a set of coupled recursions is propagated in increasing predictor order ([9] is a continuous-time version of [8]). The new algorithms of this paper can be viewed as *split* versions of those of [8], analogous to the split Levinson and Schur algorithms of [10] being split versions of the classical Levinson and Schur algorithms. Alternately, they may be viewed as analogues of the split algorithms of [10], applied not to symmetric Toeplitz systems, but to arbitrary Toeplitz-plus-Hankel systems.

The heart of the new algorithms is a four-term recurrence that generalizes the three-term recurrences of [10] to Toeplitz-plus-Hankel matrices. This recurrence requires two multiplications per update, which is half the number required by the algorithms of [7]-[9]. This is analogous to the 50% savings in multiplications for the split algorithms of [10] over the classical Levinson and Schur algorithms. To save space we refer to the new algorithms as split algorithms, rather than analogues of split algorithms, in the sequel.

In Section II the basic four-term recurrence for the new split Levinson algorithm is derived. In Section III the computation of generalized potentials using an "inner product" expression is shown; this and the four-term recurrence constitute the new split Levinson algorithm. In Section IV a new split Schur algorithm is derived; this avoids the "inner product" expression required by the split Levinson algorithm. Section V shows how the new split algorithms are used to solve arbitrary Toeplitz-plus-Hankel systems. Section

VI discusses how the new algorithms are related to previous split algorithms in special cases. Section VII concludes by summarizing and noting current research in progress on multi-dimensional versions of these algorithms.

II. DERIVATION OF THE FOUR-TERM RECURRENCE

A. The Basic Problem

In Sections II-IV we consider the solution of the Toeplitz-plus-Hankel system

$$\begin{bmatrix} 1 + k_{-i,-i} & \cdots & k_{i,-i} \\ \vdots & \vdots & \vdots \\ k_{-i,0} & \cdots & k_{i,0} \\ \vdots & \vdots & \vdots \\ k_{-i,i} & \cdots & 1 + k_{i,i} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ h_{-i,-(i-1)} & h_{i,-(i-1)} \\ \vdots & \vdots \\ h_{-i,i-1} & h_{i,i-1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} S_{-i,-i} & S_{i,-i} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ S_{-i,i} & S_{i,i} \end{bmatrix} \quad (1)$$

where the $S_{\pm i, \pm i}$ are defined from the $\{k_{i,j}\}$ and $\{h_{i,j}\}$ in (15) below, and the ij th element of the system matrix has the form

$$k_{i,j} = k_1(i-j) + k_2(i+j) \quad (2)$$

for arbitrary functions $k_1(\cdot)$ and $k_2(\cdot)$. Note in particular that the system matrix need be neither symmetric nor persymmetric; the only requirement is that all of the central submatrices be nonsingular.

Updating (1) from i to $i+1$ increases the size of the matrix by *two*; this requires two updates, and requires $k_{i/2,j/2}$ be defined at *half-integer* values $(i/2, j/2)$. If $i/2 + j/2$ is not an integer, let $k_{i/2,j/2} = 0$; if $i/2 + j/2$ is an integer, assign $k_{i/2,j/2}$ such that the matrix with ij th coordinate $k_{i/2,j/2}$ is Toeplitz-plus-Hankel. If $k_{i,j}$ is specified by the form (2), this can be done easily by inserting the half-integer values in the functions $k_1(\cdot)$ and $k_2(\cdot)$ (note that the arguments will always be integers); if only the matrix (1) is given, see Section V.

Omitting the first and last rows of (1) allows it to be rewritten as

$$0 = k_{i,j} + h_{i,j} + \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,j}; \quad 0 = k_{-i,j} + h_{-i,j} + \sum_{n=-(i-1)}^{i-1} h_{-i,n} k_{n,j}, \quad -(i-1) \leq j \leq i-1. \quad (3)$$

Now define the *interpolated* system of (3) as

$$0 = k_{i+1/2,j+1/2} + h_{i+1/2,j+1/2} + \sum_{n=-(i-1/2)}^{i-1/2} h_{i+1/2,n} k_{n,j+1/2} \quad -(i-1) \leq j \leq i-1. \quad (4)$$

and similarly for $-i-1/2$. The interpolated systems for various orders are auxiliary systems of Toeplitz-plus-Hankel systems that are solved along with (3) by the algorithms to follow. This artifice is necessary in order to obtain split algorithms solving *nested* systems (see Section VI).

B. Derivation of Four-Term Recurrence for $h_{i,j}$

To make the derivation easier to follow, we consider only positive i . Define the discrete wave operator Δ of a function $f_{i,j}$ as

$$\Delta f_{i,j} = f_{i+1/2,j} + f_{i-1/2,j} - f_{i,j+1/2} - f_{i,j-1/2} \quad (5)$$

Δ is the discrete version of the continuous operator $(\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial s^2})$. Note that the Toeplitz-plus-Hankel structure (2) is *equivalent* to

$$\Delta k_{i,j} = 0; \quad \text{for integer } i+j. \quad (6)$$

Apply the operator Δ to (3) by writing (3) with i replaced with $i \pm 1/2$, and then j replaced with $j \pm 1/2$, and then adding and subtracting (4) appropriately. Using (6) and the definition (5) gives

$$\begin{aligned} 0 = & \Delta h_{i,j} + \sum_{n=-(i-3/2)}^{i-3/2} \Delta h_{i,n} k_{n,j} + h_{i+1/2,i-1/2} k_{i-1/2,j} + h_{i+1/2,-(i-1/2)} k_{-(i-1/2),j} \\ & - \sum_{n=-(i-3/2)}^{i-3/2} (h_{i,n+1/2} (k_{n+1/2,j+1/2} - k_{n,j}) + h_{i,n-1/2} (k_{n-1/2,j-1/2} - k_{n,j})) \\ & - h_{i,i-1} k_{i-1,j-1/2} - h_{i,-(i-1)} k_{-(i-1),j+1/2} \quad -(i-3/2) \leq j \leq i-3/2 \end{aligned} \quad (7)$$

The first sum in (7) has the desired form for the argument to follow; the second sum and the two extra terms following each sum are all corrections to the first sum. Note that (7)

only holds for $-(i - 3/2) \leq j \leq i - 3/2$, since in deriving (7) we used (4) with i replaced with $i - 1/2$.

The second sum in (7) can be simplified using (6). Changing the summation variable from n to $n + 1$ in the second term shows that

$$\begin{aligned}
& - \sum_{n=-(i-3/2)}^{i-3/2} (h_{i,n+1/2}(k_{n+1/2,j+1/2} - k_{n,j}) + h_{i,n-1/2}(k_{n-1/2,j-1/2} - k_{n,j})) \\
& = \sum_{n=-(i-1/2)}^{i-3/2} h_{i,n+1/2} \Delta k_{n+1/2,j} + h_{i,i-1}(k_{i-1,j-1/2} - k_{i-1/2,j}) + h_{i,-(i-1)}(k_{-(i-1),j+1/2} - k_{-(i-1/2),j}).
\end{aligned} \tag{8}$$

The sum in (8) vanishes by (6). Substituting (8) into (7) and collecting the extra terms on the left side results in

$$0 = V_i^1 k_{i-1/2,j} + V_i^2 k_{-(i-1/2),j} + \Delta h_{i,j} + \sum_{n=-(i-3/2)}^{i-3/2} \Delta h_{i,n} k_{n,j} \tag{9}$$

where we have defined the *potentials* (see [11] for a discussion of this term)

$$V_i^1 = h_{i+1/2,i-1/2} - h_{i,i-1}; \quad V_i^2 = h_{i+1/2,-(i-1/2)} - h_{i,-(i-1)}. \tag{10}$$

Equation (9) has the same form as (4), with a different left side. To see this, write (4) with $i + 1/2$ replaced with $i - 1/2$ and $-(i - 1/2)$, multiply by V_i^1 and V_i^2 , respectively, and add. This gives

$$V_i^1 k_{i-1/2,j} + V_i^2 k_{-(i-1/2),j} = V_i^1 h_{i-1/2,j} + V_i^2 h_{-(i-1/2),j} + \sum_{n=-(i-3/2)}^{i-3/2} (V_i^1 h_{i-1/2,n} + V_i^2 h_{-(i-1/2),n}) k_{n,j} \tag{11}$$

C. Basic Four-Term Recurrence for $h_{i,j}$

Since $k_{i,j}$ is nonsingular by assumption, the solution $V_i^1 h_{i-1/2,j} + V_i^2 h_{-(i-1/2),j}$ to (11) must be unique. Comparing (10) and (11), this implies that

$$\Delta h_{i,j} = V_i^1 h_{i-1/2,j} + V_i^2 h_{-(i-1/2),j} \quad -(i - 3/2) \leq j \leq i - 3/2 \tag{12}$$

which can be written as

$$h_{i+1/2,j} = h_{i,j+1/2} + h_{i,j-1/2} + (V_i^1 - 1)h_{i-1/2,j} + V_i^2 h_{-(i-1/2),j} \quad -(i-3/2) \leq j \leq i-3/2 \quad (13)$$

Equation (13) is the four-term recurrence that is the heart of the new algorithms. It is analogous to the three-term recurrence on which the split algorithms of [10] are based, although there are some differences (see Section VI). Although we have treated i as positive throughout this derivation, (13) also holds for negative i and $-(|i| - 3/2) \leq j \leq |i| - 3/2$.

III. NEW SPLIT LEVINSON ALGORITHM

The four-term recurrence (13) can be propagated in increasing $|i|$ and $-(|i| - 3/2) \leq j \leq |i| - 3/2$. Note that for i an integer/half-integer, j will take half-integer/integer values, respectively. However, since (13) does not hold for $j = \pm(i-1)$, we must update $h_{i,\pm(i-1)}$ using (10), and similarly for $h_{-i,\pm(i-1)}$. Also, both (10) and (13) require V_i^1 and V_i^2 to be supplied separately, computed from $k_{i,j}$; note that (10) cannot be used to compute V_i^1 and V_i^2 , since (10) is needed to update $h_{\pm i,\pm(i-1)}$. We now show how V_i^1 and V_i^2 can be computed from previously computed $h_{i,j}$ and $k_{i,j}$.

A. Computation of V_i^1 and V_i^2

Setting $j = i - 1$ in (3) and (4) gives

$$h_{i+1/2,i-1/2} = -k_{i+1/2,i-1/2} - \sum_{n=-(i-1/2)}^{i-1/2} h_{i+1/2,n} k_{n,i-1/2}; \quad h_{i,i-1} = -k_{i,i-1} - \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,i-1}. \quad (14)$$

The second equation requires only $k_{i,j}$ (known) and $h_{i,j}$ (from the previous recursion); however, the first requires $h_{i+1/2,j}$, which has not yet been computed. Substituting (13) into the first equation and a considerable amount of algebra results in the following. Define the *Schur variables*

$$S_{i,j} = \delta_{i,j} + k_{i,j} + \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,j}, \quad j = \pm i \quad (15)$$

where $\delta_{i,j} = 1$ if $i = j$ and is zero otherwise. Note that $S_{i,j}$ can be computed from known

$k_{i,j}$ and $h_{i,j}$. Then it may be shown that

$$\begin{bmatrix} S_{i-1/2,i-1/2} & S_{-(i-1/2),i-1/2} \\ S_{i-1/2,-(i-1/2)} & S_{-(i-1/2),-(i-1/2)} \end{bmatrix} \begin{bmatrix} V_i^1 \\ V_i^2 \end{bmatrix} = \begin{bmatrix} S_{i-1/2,i-1/2} - S_{i,i} \\ S_{i-1/2,-(i-1/2)} - S_{i,-i} \end{bmatrix} \quad (16)$$

The existence of a unique solution to (16) is proved in Section V below. The closed-form solution of (16) is

$$V_i^1 = (S_{-(i-1/2),-(i-1/2)}(S_{i-1/2,i-1/2} - S_{i,i}) - S_{-(i-1/2),i-1/2}(S_{i-1/2,-(i-1/2)} - S_{i,-i})) / DET \quad (17a)$$

$$V_i^2 = (S_{i-1/2,i-1/2}(S_{i-1/2,-(i-1/2)} - S_{i,-i}) - S_{i-1/2,-(i-1/2)}(S_{i-1/2,i-1/2} - S_{i,i})) / DET \quad (17b)$$

$$DET = S_{i-1/2,i-1/2}S_{-(i-1/2),-(i-1/2)} - S_{i-1/2,-(i-1/2)}S_{-(i-1/2),i-1/2}. \quad (17c)$$

B. New Split Levinson Algorithm

$$\text{Initialization : } h_{\pm 1,0} = -k_{\pm 1,0}/(1 + k_{0,0}) \quad (18)$$

Computation of V_i^1, V_i^2 : Compute $S_{i,\pm i}$ from $k_{i,j}$ (known) and $h_{i,j}$ (from previous recursion) using (15). Compute V_i^1 and V_i^2 from $S_{i,\pm i}$ and $S_{i-1,\pm(i-1)}$ using (17).

Update $h_{i,j}$: Compute $h_{\pm(i+1/2),\pm(i-1/2)}$ using (from (10))

$$h_{i+1/2,i-1/2} = h_{i,i-1} + V_i^1; \quad h_{i+1/2,-(i-1/2)} = h_{i,-(i-1)} + V_i^2 \quad (19a)$$

$$h_{-(i+1/2),i-1/2} = h_{-i,i-1} + V_{-i}^1; \quad h_{-(i+1/2),-(i-1/2)} = h_{-i,-(i-1)} + V_{-i}^2. \quad (19b)$$

Compute $h_{i+1/2,j}$, $-(i-3/2) \leq j \leq (i-3/2)$ using (13). Compute $h_{-(i+1/2),j}$ by writing (12) as

$$h_{-(i+1/2),j} = h_{-i,j+1/2} + h_{-i,j-1/2} + (V_{-i}^1 - 1)h_{-(i-1/2),j} + V_{-i}^2 h_{i-1/2,j} \quad (20)$$

At this point the recursion is complete. The computed $h_{i,j}$ for integer/ half-integer i and j solve the original system (3)/interpolated system (4), respectively; note that two recursions are needed to increase the size of the system (3) by two (i.e., update i to $i+1$).

The heart of the algorithm is the four-term recurrence (13), which requires $2i - 3$ multiplications to update $h_{i,j}$. The fast algorithms of [7]-[9] require roughly $4i$ multiplications to update $h_{i,j}$. There is a redundancy in the computations of [7]-[9] similar to that in the classical Levinson and Schur algorithms; the savings of roughly 50% is analogous to the savings in the split Levinson and Schur algorithms of [10] over the classical algorithms.

This algorithm differs from the split Levinson algorithm of [10] in two other respects. First, the non-symmetric Toeplitz-plus-Hankel system matrix requires *four* sequences $V_{\pm i}^1$ and $V_{\pm i}^2$ of potentials and the four-term recurrence (13). The symmetric Toeplitz system matrix solved by the split Levinson algorithm of [10] requires only one sequence of potentials and a three-term recurrence. Second, the split Levinson algorithm of [10] propagates not $h_{i,j}$ but $h_{i,j} + h_{i,-j}$; this is more efficient for symmetric Toeplitz matrices, but requires recovery of $h_{i,j}$ from $h_{i,j} + h_{i,-j}$ at termination.

IV. NEW SPLIT SCHUR ALGORITHM

The "inner product" (15) computation requires i multiplications; since it is not parallelizable, it is a computational bottleneck, just as in the classical Levinson algorithm. For this reason, we now derive a new split Schur-type algorithm for arbitrary Toeplitz-plus-Hankel matrices. This algorithm can be propagated in parallel with the split Levinson algorithm derived above; this avoids the computational bottleneck (15). The same idea was used for the classical Schur and Levinson algorithms in [12].

The first step is to show that the forward prediction error filter satisfies the four-term recurrence (13). From this, we show that the $S_{i,j}$ defined in (15) (now for all $j \geq i$) also satisfy (13). This implies that (13), initialized using $k_{i,j}$, can be used to compute V_i^1 and V_i^2 quickly.

A. Four-Term Recurrence for $S_{i,j}$

Define $\phi_{i,j}$ by

$$\phi_{i,j} = \delta_{i,j} + h_{i,j} \quad (21)$$

Clearly $\phi_{i,j}$ satisfies (13) for $-(i - 3/2) \leq j \leq i - 3/2$ since $\phi_{i,j} = h_{i,j}$ for these values. At $j = \pm(i - 1/2)$ or $\pm(i + 1/2)$ $\phi_{i,j}$ satisfies (13), since this reduces to (10). And for

$|j| \geq i + 3/2$ (13) reduces to $0 = 0$. Hence (13) with $h_{i,j}$ replaced with $\phi_{i,j}$ is true for *all* i an integer/half-integer and j a half-integer/integer:

$$\phi_{i+1/2,j} = \phi_{i,j+1/2} + \phi_{i,j-1/2} + (V_i^1 - 1)\phi_{i-1/2,j} + V_i^2\phi_{-(i-1/2),j} \quad (22)$$

Next, extend the definition $S_{i,j}$ in (15) to all integers and half-integers i and j such that $i + j$ is an integer:

$$S_{i,j} = \delta_{i,j} + k_{i,j} + \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,j}; \quad S_{i+1/2,j+1/2} = \delta_{i,j} + k_{i+1/2,j+1/2} + \sum_{n=-(i-1/2)}^{i-1/2} h_{i+1/2,n} k_{n,j+1/2} \quad (23)$$

From (3) and (4) $S_{i,j} = 0$ for $-(i-1) \leq j \leq i-1$. Substituting (2) and (21) in (23) gives

$$\begin{aligned} S_{i,j} &= \sum (\delta_{i,n} + h_{i,n})(\delta_{n,j} + k_{n,j}) = \sum \phi_{i,n}(\delta_{n,j} + k_1(j-n) + k_2(j+n)) \\ &= \phi_{i,j} + \phi_{i,j} * k_1(j) + \phi_{i,-j} * k_2(j) \end{aligned} \quad (24)$$

where $*$ denotes a convolution in j .

Since (22) is linear in functions of j , it may be convolved with $k_1(j)$. Note that (22) still holds if j is replaced with $-j$, and convolve this with $k_2(j)$. Adding (22) to the convolution of (22) with $k_1(j)$ and the convolution of the time-reversal of (22) with $k_2(j)$ and using (24) shows that

$$S_{i+1/2,j} = S_{i,j+1/2} + S_{i,j-1/2} + (V_i^1 - 1)S_{i-1/2,j} + V_i^2 S_{-(i-1/2),j} \quad (25)$$

so that $S_{i,j}$ also satisfies the four-term recurrence (13). Equation (25) can also be derived by taking the z -transform in j of (22), noting that the result is unaffected if z is replaced with $1/z$, multiplying by the z -transforms of $k_1(j)$ and $k_2(j)$, and adding.

B. New Split Schur Algorithm

$$\text{Initialization : } S_{0,j} = k_{0,j}; \quad S_{\pm 1/2,j+1/2} = k_{\pm 1/2,j+1/2} \quad (26)$$

Note $k_{0,m}$ and $k_{1/2,n+1/2}$ for integer m and half-integer $n + 1/2$ uniquely determines $k_{i,j}$ for *all* i, j , $i + j$ an integer, using (6).

Computation of V_i^1, V_i^2 : Compute V_i^1 and V_i^2 from $S_{i,\pm i}$ and $S_{i-1/2,\pm(i-1/2)}$ using (17). Similar equations are used to compute V_{-i}^1 and V_{-i}^2 .

Update $S_{i,j}, |j| \geq i$ using (25).

At this point the recursion is complete. The split Schur algorithm can be run in parallel with the split Levinson algorithm, supplying the potentials $V_{\pm i}^1$ and $V_{\pm i}^2$ while bypassing the "inner product" computation (15) ((17) is still necessary), as suggested in [12] for the classical algorithms.

If the original system (3) is a discretization of an integral equation, then $S_{i,j} \ll 1$ and the $\delta_{i,j}$ in (15) dominates the other terms if $i = j$. In this case the solution to (16) is simply

$$V_i^1 = S_{i-1,i-1} - S_{i,i}; \quad V_i^2 = S_{i-1,-(i-1)} - S_{i,-i} \quad (27)$$

which replaces the more complicated (17).

V. SOLUTION OF ARBITRARY TOEPLITZ-PLUS-HANKEL SYSTEMS

The split algorithms above solve the systems (3) and (4); hence they also solve (1) with $S_{\pm i,\pm i}$ defined as in (15). We now consider the general problem

$$\begin{bmatrix} 1 + k_{-i,-i} & \cdots & k_{i,-i} \\ \vdots & \ddots & \vdots \\ k_{-i,i} & \cdots & 1 + k_{i,i} \end{bmatrix} \begin{bmatrix} x_{-i} \\ \vdots \\ x_i \end{bmatrix} = \begin{bmatrix} b_{-i} \\ \vdots \\ b_i \end{bmatrix} \quad (28)$$

where the right side is now arbitrary.

Define $\{c_j, -i \leq j \leq i\}$ recursively as follows. Let $c_{\pm j}$ be the solution to the 2×2 system

$$\begin{bmatrix} S_{-j,-j} & S_{j,-j} \\ S_{-j,j} & S_{j,j} \end{bmatrix} \begin{bmatrix} c_{-j} \\ c_j \end{bmatrix} = \begin{bmatrix} b_{-j} - \sum_{n=-(j-1)}^{j-1} c_n S_{n,-j} \\ b_j - \sum_{n=-(j-1)}^{j-1} c_n S_{n,j} \end{bmatrix}. \quad (29)$$

Then the solution to (28) is given by

$$x_j = \sum_{n=-i}^i c_n \phi_{n,j}, \quad -i \leq j \leq i \quad (30)$$

where $\phi_{i,j}$ is defined in (21) and $h_{i,j}$ is defined to be zero for $|i| < |j|$. These equations may be derived easily by taking linear combinations (weighted by the c_{\pm}) of the columns

of (1) for increasing i and equating to (28). Note how this relies on the split algorithms solving *nested* systems of equations as i increases.

We note here that the 2×2 systems (16) and (29) have unique solutions if the central submatrices of the system matrix (1) are nonsingular. To see this, suppose that the 2×2 system matrix in (16) and (29) is singular. Then the second column is a multiple (say m) of the first column, and the column vector $[1, \dots, (h_{-i,j} - mh_{i,j}), \dots, -m]^T$ solves the homogeneous system associated with (1), which is impossible as long as the system matrix in (1) is nonsingular.

If the system matrix is specified by functions $k_1(\cdot)$ and $k_2(\cdot)$ as in (2) (and [7]), then the initialization (26) for the split Schur algorithm is accomplished using (2) directly (note the arguments are always integers). However, if the matrix (1) is given directly, then $k_{\pm 1/2, j+1/2}$ must be interpolated from the given values $k_{0,j}$ and $k_{\pm 1,j}$. From (6), these can be recursively computed as needed in the split Schur algorithm using

$$k_{\pm 1/2, j+1/2} = k_{0,j} + k_{\pm 1,j} - k_{\pm 1/2, j-1/2}; \quad k_{\pm 1/2, 1/2} = k_{\pm 1/2, -1/2} = (1 + k_{0,0} + k_{\pm 1,0})/2 \quad (31)$$

VI. RELATION WITH PREVIOUS SPLIT ALGORITHMS

A. Relation to the Split Algorithms of [10]

To show how the new algorithms reduce to the split algorithms of [10], we first consider the class of Toeplitz-plus-Hankel matrices such that $k_{i,j} = k_{-i,-j}$. In terms of (2) both $k_1(\cdot)$ and $k_2(\cdot)$ are even functions; note that covariance functions of time-reversible random processes have this property. The set of *centrosymmetric* matrices (matrices that are both persymmetric $k_{i,j} = k_{-j,-i}$ and symmetric $k_{i,j} = k_{j,i}$) is a subset of this class. From (3) $h_{i,j} = h_{-i,-j}$, from (15) $S_{i,j} = S_{-i,-j}$, and from (17) $V_i^1 = V_{-i}^2$, and $V_i^2 = V_{-i}^1$. Hence the computations for $i < 0$ can all be dispensed with.

We can go further. Defining

$$a_{i,j} = h_{i,j} + h_{i,-j}; \quad e_{i,j} = S_{i,j} + S_{i,-j}; \quad V_i = V_i^1 + V_i^2 \quad (32)$$

replacing j with $-j$ in (12) and (25), and adding to (12) and (25) respectively results in

$$\Delta a_{i,j} = V_i a_{i-1,j}; \quad \Delta e_{i,j} = V_i e_{i-1,j}. \quad (33)$$

Adding (16a) and (16b) allows V_i to be computed from $e_{i,j}$ by

$$V_i = (e_{i-1,i-1} - e_{i,i})/e_{i-1,i-1}. \quad (34)$$

From (3) and (32) $a_{i,j}$ is the solution to

$$k_{i,j} + k_{i,-j} = a_{i,j} + \sum_{n=-(i-1)}^{i-1} a_{i,n} k_{n,j}. \quad (35)$$

The solution to (35) can be recursively computed using the *three-term* recurrences (33), along with (34). These equations have virtually the same form as the split algorithms of [10], *even though $k_{i,j}$ is not Toeplitz*.

To see what is happening here, use (2) to rewrite the left side of (35) as

$$k_{i,j} + k_{i,-j} = k_1(i-j) + k_2(i+j) + k_1(i+j) + k_2(i-j) = k(i-j) + k(i+j) \quad (36)$$

where $k(i) = k_1(i) + k_2(i)$. From (32) $a_{i,j} = a_{i,-j}$, and the right side of (35) can be rewritten using this and (36), yielding

$$k(i-j) + k(i+j) = a_{i,j} + \sum_{n=0}^{i-1} a_{i,n} (k(n-j) + k(n+j)). \quad (37)$$

Equation (37) is in fact the symmetric Toeplitz system solved by the split algorithms of [10], after shifting from a one-sided to a two-sided interval. This shows how these algorithms are related to the algorithms of this paper. Note that the split algorithms of [10] propagate $a_{i,j}$, not $h_{i,j}$; $h_{i,j}$ must be computed from $a_{i,j}$ at the end.

If the system matrix (1) is merely symmetric, a more subtle simplification is possible. In this case, the block-Toeplitz reformulation of [7] becomes a centrosymmetric block-Toeplitz problem, and the results of Section VI of [13] can be used to derive a three-term *matrix* recurrence similar in form to (13) and (20) combined, except that $V_{-i}^2 = V_i^2$. However, this recurrence does not propagate the $h_{\pm i,j}$ directly, but weighted combinations of them, and it requires as many multiplications as the algorithm of this paper (which also works for nonsymmetric matrices). It is more efficient in that it requires only three functions, instead of four, to characterize the inverse of the system matrix (1); this is

reasonable since symmetry requires $k_1(\cdot)$ in (2) to be an even function, removing a degree of freedom.

B. An Alternative Algorithm for Non-Nested Matrices

If the problem (1) is modified so that the system matrices of different orders are no longer nested, the algorithm takes a slightly different form. Consider the system

$$\begin{bmatrix} k_{1,1} & \cdots & k_{n+2,1} \\ \vdots & \vdots & \vdots \\ k_{1,n+2} & \cdots & k_{n+2,n+2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_1^n & y_n^n \\ \vdots & \vdots \\ x_n^n & y_1^n \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} S_0^n & T_0^n \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ S_{n+1}^n & T_{n+1}^n \end{bmatrix} \quad (38)$$

where $k_{i,j}$ is now defined by

$$k_{i,j} = k_1(i-j) + k_2(i+j-(n+3)) \quad (39)$$

and S_i^n and T_i^n are defined as (compare to (15))

$$S_i^n = \sum_{j=0}^n x_j^n k_{j+1,i+1}; \quad T_i^n = \sum_{j=0}^n y_j^n k_{n+2-j,i+1}; \quad x_0^n = y_0^n = 1. \quad (40)$$

Defining the polynomials

$$X_n(z) = \sum_{j=0}^n x_j^n z^j; \quad Y_n(z) = \sum_{j=0}^n y_j^n z^j; \quad R(z) = \sum_{j=-\infty}^{\infty} k_1(j) z^j; \quad H(z) = \sum_{j=-\infty}^{\infty} k_2(j) z^j \quad (41)$$

the system (38) can be written in polynomial form as

$$R(1/z)X_n(z) + z^{n+1}H(z)X_n(1/z) = \dots + S_0^n + S_{n+1}^n z^{n+1} + \dots \quad (42a)$$

$$R(1/z)z^{n+1}Y_n(1/z) + H(z)Y_n(z) = \dots + T_0^n + T_{n+1}^n z^{n+1} + \dots \quad (42b)$$

where the ellipses indicate terms of lower and higher order in Laurent series.

Knowing the form of the four-term recurrence, writing (42) for n , $n+1$, and $n-1$, and adding and subtracting appropriately gives

$$R(1/z) (X_{n+1}(z) - (z+1)X_n(z) - zV_n^1 X_{n-1}(z) - zV_n^2 z^n Y_{n-1}(1/z))$$

$$\begin{aligned}
& +H(z) (z^{n+2}X_{n+1}(1/z) - z^{n+2}(1/z+1)X_n(1/z) - z^{n+2}(1/z)V_n^1X_{n-1}(1/z)) \\
& = \dots + S_0^{n+1} + S_{n+2}^{n+1}z^{n+2} + \dots
\end{aligned} \tag{43}$$

provided that V_n^1 and V_n^2 are chosen such that (compare to (16))

$$\begin{bmatrix} S_0^{n-1} & T_0^{n-1} \\ S_n^{n-1} & T_n^{n-1} \end{bmatrix} \begin{bmatrix} V_n^1 \\ V_n^2 \end{bmatrix} = \begin{bmatrix} -S_0^n \\ -S_{n+1}^n \end{bmatrix} \tag{44}$$

As long as the system matrix (38) is invertible, the expression in parentheses in (43) must be zero. Equations (43) and (44) define a four-term recurrence for the solutions to (38). Proceeding as before, analogues of split Levinson and Schur algorithms may be derived.

This algorithm avoids the interpolated system and half-integer recursions of the previous algorithms. However, it does not save any computation. More importantly, (38) and (39) do not define a *nested* set of system matrices in increasing order n : the ij th element of the system matrix changes with order n (see (39)). Hence this algorithm is not useful for *updating* problems, in which the size of a Toeplitz-plus-Hankel system is enlarged by augmenting the system matrix around its edges; this type of problem is common in linear prediction. The solution of an arbitrary Toeplitz-plus-Hankel system also becomes more complex than (29)-(30).

A *nested* system of equations in increasing n can be defined from (38) and (39) by making the substitution $i' = i - (n+3)/2$, $j' = j - (n+3)/2$. This alters (38) and (39) to (1) and (2), respectively; however, for n even it requires that the interpolated system (4) be defined. This leads back to the previous algorithm.

Although the derivation (43) is simpler than that of Section II, it requires prior knowledge of the form of the recurrence. The derivation of Section II *derives* the form of the recurrence directly, and shows that the Toeplitz-plus-Hankel form, rather than the purely Toeplitz form, is fundamental to the split-like recurrences. It also shows that matrices with structure defined *implicitly* (as in (6)), rather than *explicitly* (as in (39)), can have fast algorithms easily derived for them. In particular, this has led to fast algorithms for *block* Toeplitz-plus-Hankel systems of equations [14].

VII. CONCLUSION

New fast algorithms have been derived for solving arbitrary Toeplitz-plus-Hankel systems of equations. The new algorithms can be viewed as analogues of the split Levinson and Schur algorithms of [10], but applicable to a more general problem. The split Levinson algorithm recursively computes the solution using a four-term recurrence, but requires a non-parallelizable computation (15) to compute the potentials. The split Schur algorithm computes the potentials using a similar four-term recurrence; using it in parallel with the split Levinson algorithm obviates (15) and allows the same processor architecture to be used for both algorithms.

The algorithms presented in this paper have two-dimensional analogues applicable to the linear prediction problem for a two-dimensional random field [14],[15]. Unresolved issues include the numerical stability of these algorithms, optimal processor architectures for implementation, and generalization to matrices with singular submatrices.

ACKNOWLEDGMENT

The author would like to thank one of the referees for noting the significance of the applicability of these algorithms to non-symmetric matrices, and for noting the alternative algorithm in Section VI. The help of Wen-Hsien Fang in numerical testing of the new algorithms is gratefully acknowledged. This research was supported by the Air Force Office of Scientific Research under grant AFOSR-89-0017.

REFERENCES

1. A.-C. Lee, "A new autoregressive method for high-performance spectrum analysis," *J. Acoust. Soc. Am.*, vol. 86, no. 1, pp. 150-157, July 1989.
2. B. Friedlander and M. Morf, "Least-squares algorithms for adaptive linear-phase filtering," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-30, no. 3, pp. 381-389, June 1982.
3. S.M. Kay and S.L. Marple, "Spectrum analysis—a modern perspective," *Proc. IEEE*, vol. 69, no. 11, pp. 1380-1419, Nov. 1981. p. 1407.
4. W.B. Gragg and G.D. Johnson, "The Laurent-Pade table," in *Proc. IFIP Congr. 1974*. Amsterdam, the Netherlands: North-Holland, 1974, pp. 632-637.
5. J. Casti, R. Kalaba, and S. Ueno, "Source functions for an isotropically scattering

- atmosphere bounded by a specular reflector," *J. Quant. Spectrosc. Radiat. Transfer*, vol. 10, pp. 1119-1128, 1970.
6. H.H. Kagiwada and R. Kalaba, *Integral Equations via Imbedding Methods*. Reading, Mass.: Addison-Wesley, 1974.
 7. G.A. Merchant and T.W. Parks, "Efficient solution of a Toeplitz-plus Hankel coefficient matrix system of equations," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-30, no. 1, pp. 40-44, Feb. 1982.
 8. B. Friedlander and M. Morf, "Efficient inversion formulas for sums of products of Toeplitz and Hankel matrices," in *Proc. 18th Annual Allerton Conf. Comm., Cont., Comput.*, Oct. 8-10, 1980.
 9. J.N. Tsitsiklis and B.C. Levy, "Integral equations and resolvents of Toeplitz plus Hankel kernels," Tech. Report #LIDS-P-1170, Laboratory for Information and Decision Systems, M.I.T., Cambridge, Mass., Dec. 1981.
 10. P. Delsarte and Y. Genin, "On the splitting of classical algorithms in linear prediction theory," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-35, no. 5, pp. 645-653, May 1987.
 11. A.E. Yagle, "Fast algorithms for estimation and signal processing: an inverse scattering framework," to appear in *IEEE Trans. Acoust., Speech, and Sig. Proc.*, June 1989.
 12. S.Y. Kung and Y.H. Hu, "A highly concurrent algorithm and pipelined architecture for solving Toeplitz systems," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-31, no. 1, pp. 66-75, Feb. 1983.
 13. P. Delsarte and Y. Genin, "Multichannel Singular Predictor Polynomials," *IEEE Trans. Circuits and Systems*, vol. CAS-35, no. 2, pp. 190-200, Feb. 1988.
 14. A.E. Yagle and W.-H. Fang, "Discrete fast algorithms for two-dimensional linear prediction on a polar raster," submitted to *IEEE Trans. ASSP*.
 15. A.E. Yagle, "Generalized split Levinson, Schur, and lattice algorithms for three-dimensional random field estimation problems," to appear in *SIAM J. Appl. Math.*

APPENDIX C

W.-H. Fang and A.E. Yagle, "Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992.

Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster

Wen-Hsien Fang and Andrew E. Yagle
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-2122

Revised October 1990

Abstract

New generalized split Levinson and Schur algorithms for the two-dimensional linear least-squares prediction problem on a polar raster are derived. The algorithms compute the prediction filter for estimating a random field at the edge of a disk, from noisy observations inside the disk. The covariance function of the random field is assumed to have a Toeplitz-plus-Hankel structure for both its radial part and its transverse (angular) part. This assumption is valid for some types of random fields, such as isotropic random fields. The algorithms generalize the split Levinson and Schur algorithms in two ways: (1) to two dimensions; and (2) to Toeplitz-plus-Hankel covariances.

I INTRODUCTION

The problem of computing linear least-squares estimates of two-dimensional random fields from noisy observations has many applications in image processing. In particular, the two-dimensional discrete linear prediction problem is a useful formulation of problems in image smoothing and coding [1]. If the random field: (1) is defined on a rectangular lattice of points; (2) is stationary; and (3) has quarter-plane or asymmetric half-plane causality, then the two-dimensional linear prediction problem may be solved using the multichannel Levinson algorithm [2, 3] (modifications of these conditions are also possible). By exploiting the Toeplitz-block-Toeplitz structure of the covariance function of the stationary random field, this algorithm allows the linear prediction filters to be computed recursively using significantly fewer computations than direct solution of the two-dimensional discrete Wiener-Hopf equations. The multichannel Schur algorithm computes the reflection coefficient matrices from the covariance function; propagating it in parallel with the Levinson algorithm saves even more computation.

In tomographic imaging problems solved by filtered back-projection [4], and in spotlight synthetic aperture radar [5], data are collected on a polar raster of points, rather than on a rectangular lattice. Although such data can be interpolated onto a rectangular lattice, this is necessarily inexact; it also affects the covariance function. For example, the covariance of an isotropic random field on a rectangular lattice is a Toeplitz function of the ordinates and abscissae, while on a polar raster it is a Toeplitz-plus-Hankel function of the radii. For smoothing noisy images and performing image coding for images defined on a polar raster, it is clearly desirable to develop analogues of the multichannel Levinson and Schur algorithms applicable to discrete random fields defined on a polar raster.

This paper develops these analogues. They generalize previous results in three ways: (1) the random field is defined on a polar raster; (2) the random field is not required to be stationary; rather, its covariance must have Toeplitz-plus-Hankel structure in both the radial and transverse directions (some important cases of such random fields are noted in Section IV); and (3) the quarter-plane or asymmetric half plane causality assumption is replaced by a more natural causality defined in the radial direction only. The

prediction filters estimate the random field at a given point using observations from all points of smaller radius.

Three other features are worth noting here. First, the algorithms are generalized three-term recursions, similar in structure to the split algorithms [6, 7]. The one-dimensional split algorithm recursions require only half as many multiplications as the two-component lattice recursions of the Levinson and Schur algorithms. Our two-dimensional algorithms are similarly computationally efficient, which is important in two-dimensional signal processing. Second, the *smoothing* filters for estimating the random field from observations at points of smaller *and greater* radii can be easily computed [8] from the prediction filters using a discrete multi-dimensional generalization of the application of the Bellman-Siebert-Krein identity to the one-dimensional smoothing problem in [9]. Indeed, the new two-dimensional algorithms of this paper are applied to arbitrary Toeplitz-plus-Hankel-block-Toeplitz-plus-Hankel systems in Section V.

Finally, we note that similar ideas have been applied to continuous-parameter isotropic [10] and homogeneous [11] random fields, and to random fields with more general Toeplitz-plus-Hankel structure in [12] and [13]; this paper can be viewed as a discrete version of the results of [13]. Although the continuous algorithm can always be discretized, an inherently discrete algorithm can be expected to perform better on a computer; there are minor yet significant differences between the results of this paper and the continuous results of [13] (see Section IV). Also, in some problems the data are sampled, or only taken at discrete points. These facts motivate us to develop a discrete counterpart of the continuous algorithms. We also note that the one-dimensional version of this algorithm has been presented in [14], and that a summary of the results of this paper was presented in [15].

This paper is organized as follows. In Section II, the two-dimensional analogue of the discrete split Levinson recurrence for the linear prediction problem on a polar raster is derived. The derivation is based on the assumption that both the radial part and the transverse part of the covariance have Toeplitz-plus-Hankel structure. Section III derives a corresponding Schur algorithm, to be propagated in parallel with

the Levinson algorithm. Some examples of random fields with covariances having Toeplitz-plus-Hankel structure are discussed in Section IV, and comparison with the results of [13] are made. In Section V, the computational complexity of the proposed algorithm is evaluated, and compared with other algorithms. The solution to a general Toeplitz-plus-Hankel block Toeplitz-plus-Hankel system of equations is also developed. Section VI concludes with a summary.

II DERIVATION OF THE LEVINSON-LIKE RECURRENCE

A. Basic Problem

The problem considered is as follows. Given noisy observations $\{y_{i,N}\}$ of a zero-mean real-valued discrete random field $\{x_{i,N}\}$ at the points (i, N) of a polar raster on a disk, compute the linear least-squares predictions of $x_{i,N}$ for all points on the edge of the disk using all the data inside the disk. Here i is an integer radius from the origin, and N is the integer index of the argument (angle); if there are M points distributed on the circle of any radius, then (i, N) is the point at radius i and angle $2\pi N/M$.

The observations $\{y_{i,N}\}$ are related to the field $\{x_{i,N}\}$ by $y_{i,N} = x_{i,N} + v_{i,N}$, where $\{v_{i,N}\}$ is a zero-mean discrete white noise field with unit power, and $\{x_{i,N}\}$ and $\{v_{i,N}\}$ are uncorrelated (white noise with arbitrary power σ^2 can be easily handled by scaling). The covariance of $\{x_{i,N}\}$ is

$$K(i, N_1; j, N_2) \triangleq E[x_{i,N_1} x_{j,N_2}] \quad (1)$$

which is assumed to be a non-negative definite function with Toeplitz-plus-Hankel structure in both arguments (this is defined precisely in (13) and (14) below). Although an actual covariance would also be symmetric function, symmetry in (1) is not required by the algorithms to follow; this permits their application to general Toeplitz-plus-Hankel block Toeplitz-plus-Hankel systems in Section V.

The estimates of $x_{i,N}$ at the edge of the disk are computed from the observations $\{y_{i,N}\}$ using

$$\hat{x}_{i,N_1} = \sum_{j=0}^{i-1} \sum_{N_2=1}^M h(i, N_1; j, N_2) y_{j,N_2} \quad (2)$$

By the orthogonality principle of linear prediction, the optimal prediction filters $h(i, N_1; j, N_2)$ are

computed from the covariance $K(i, N_1; j, N_2)$ by solving the two-dimensional discrete Wiener-Hopf equation

$$K(i, N_1; j, N_2) = h(i, N_1; j, N_2) + \sum_{n=0}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) K(n, N_3; j, N_2) \quad (3)$$

for all $0 \leq j \leq i-1$ and $1 \leq N_1, N_2 \leq M$.

The goal of this paper is to derive fast algorithms for solving (3) for $h(i, N_1; j, N_2)$ when $K(i, N_1; j, N_2)$ has Toeplitz-plus-Hankel block Toeplitz-plus-Hankel structure.

For convenience in the derivation, we solve not (3) but the system of equations

$$K(i, N_1; j, N_2) = h(i, N_1; j, N_2) + \sum_{n=-(i-1)}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) K(n, N_3; j, N_2) \quad (4)$$

for all $-(i-1) \leq j \leq i-1$ and $1 \leq N_1, N_2 \leq M$. This modified system (4) is motivated by noting that the continuous-parameter two-dimensional Wiener-Hopf integral equation

$$\begin{aligned} K(x, y) &= h(x, y) + \int_{|z| \leq |x|} h(x, z) K(z, y) dz \\ &= h(x, y) + \int_0^{|x|} \int_0^{2\pi} h(x, |z|\theta) K(|z|\theta, y) |z| d\theta d|z|, \quad x, y, z \in \mathbb{R}^2, |y| \leq |x| \end{aligned} \quad (5)$$

discretizes into

$$K'(i, N_1; j, N_2) = h'(i, N_1; j, N_2) + \sum_{n=0}^{i-1} \sum_{N_3=1}^M h'(i, N_1; n, N_3) K'(n, N_3; j, N_2) n \quad (6)$$

where the radial weighting factor n in (6) reflects $|z|$ in (5). If we let

$$h(i, N_1; j, N_2) = \frac{\sqrt{ij}}{2} h'(i, N_1; j, N_2) = \frac{\sqrt{ij}}{2} h'(i, N_1; -j, N_2 + M/2) \quad (7)$$

$$K(i, N_1; j, N_2) = \frac{\sqrt{ij}}{2} K'(i, N_1; j, N_2) = \frac{\sqrt{ij}}{2} K'(i, N_1; -j, N_2 + M/2) \quad (8)$$

then the sum in (4) is simply double the sum in (6), so that if $h(i, N_1; j, N_2)$ and $K(i, N_1; j, N_2)$ satisfy (4), then $h'(i, N_1; j, N_2)$ and $K'(i, N_1; j, N_2)$ satisfy (6). Note that the second equalities in (7) and (8) will hold on a polar raster, but are not required in (4). For convenience we continue to refer to $K(i, N_1; j, N_2)$ in (4) as the covariance function.

Similarly to the approach used in [14], we decompose the update procedure into two steps by introducing an *interpolated* (auxiliary) system. As shown in Fig. 1, between every pair of points in the same

radial direction, we insert an auxiliary point. The covariance function $K(i, N_1; j, N_2)$ is interpolated at these auxiliary points such that the Toeplitz-plus-Hankel structure (see (13),(14)) is maintained. Then the interpolated system is defined as

$$K(i, N_1; j + \frac{1}{2}, N_2) = h(i, N_1; j + \frac{1}{2}, N_2) + \sum_{n=-(i-1)}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) K(n, N_3; j + \frac{1}{2}, N_2) \quad (9)$$

for interpolation at half-integer values of j and

$$K(i + \frac{1}{2}, N_1; j + \frac{1}{2}, N_2) = h(i + \frac{1}{2}, N_1; j + \frac{1}{2}, N_2) + \sum_{n=-(i-\frac{1}{2})}^{i-\frac{1}{2}} \sum_{N_3=1}^M h(i + \frac{1}{2}, N_1; n, N_3) K(n, N_3; j + \frac{1}{2}, N_2) \quad (10)$$

for interpolation at half-integer values of i . Note that in (10) j can also take on half-integer values.

B. Derivation of the Basic Levinson-Like Recurrence

Define the discrete wave operators Δ_r and Δ_θ by

$$\Delta_r f(i, N_1; j, N_2) \triangleq f(i + \frac{1}{2}, N_1; j, N_2) + f(i - \frac{1}{2}, N_1; j, N_2) - f(i, N_1; j + \frac{1}{2}, N_2) - f(i, N_1; j - \frac{1}{2}, N_2) \quad (11)$$

$$\begin{aligned} \Delta_\theta f(i, N_1; j, N_2) &\triangleq f(i - \frac{1}{2}, N_1 + 1; j, N_2) + f(i - \frac{1}{2}, N_1 - 1; j, N_2) \\ &\quad - f(i - \frac{1}{2}, N_1; j, N_2 + 1) - f(i - \frac{1}{2}, N_1; j, N_2 - 1) \end{aligned} \quad (12)$$

where Δ_r and Δ_θ can be regarded as discrete versions of the continuous operators $(\frac{\partial^2}{\partial r_1^2} - \frac{\partial^2}{\partial r_2^2})$ and $(\frac{\partial^2}{\partial \theta_1^2} - \frac{\partial^2}{\partial \theta_2^2})$ for the radial part and transverse part, respectively. In (12) $N_1 \pm 1$ and $N_2 \pm 1$ are computed mod M , reflecting their definition as angular variables on a polar raster.

We assume that the covariance function has Toeplitz-plus-Hankel block Toeplitz-plus-Hankel structure, defined by

$$\Delta_r K(i, N_1; j, N_2) = 0 \quad (13)$$

$$\Delta_\theta K(i, N_1; j, N_2) = 0 \quad (14)$$

Applying the Laplacian operator $\Delta = \Delta_r + \Delta_\theta$ to the equation (4), we have after some algebra

$$\Delta K(i, N_1; j, N_2) = 0 = \Delta h(i, N_1; j, N_2) + \sum_{n=-(i-\frac{1}{2})}^{i-\frac{1}{2}} \sum_{N_3=1}^M \Delta h(i, N_1; n, N_3) K(n, N_3; j, N_2)$$

$$\begin{aligned}
& + \sum_{N_3=1}^M [h(i + \frac{1}{2}, N_1; i - \frac{1}{2}, N_3) - h(i, N_1; i - 1, N_3)] K(i - \frac{1}{2}, N_3; j, N_2) \\
& + \sum_{N_3=1}^M [h(i + \frac{1}{2}, N_1; -(i - \frac{1}{2}), N_3) - h(i, N_1; -(i - 1), N_3)] K(-(i - \frac{1}{2}), N_3; j, N_2) \\
& + \sum_{n=-(i-1)}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) \Delta_r K(n, N_3; j, N_2) \\
& + \sum_{n=-(i-\frac{3}{2})}^{i-\frac{3}{2}} \sum_{N_3=1}^M h(i - \frac{1}{2}, N_1; n, N_3) \Delta_\theta K(n, N_3; j, N_2) \tag{15}
\end{aligned}$$

The algebra required to derive (15) is a generalization of the algebra in [14]; the major difference is that there are no "end effect" terms in the sums over N_3 when Δ_θ applied. This is true since $h(i, N_1; j, N_2)$ is periodic with period M in N_1 and N_2 , since these indices represent angles on the polar raster.

Using (13) and (14) to note that the last two terms in (15) are zero, we note that (15) has the same form as the following linear combination:

$$\begin{aligned}
& \sum_{N_3=1}^M [V_i^+(N_1, N_3) K(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3) K(-(i - \frac{1}{2}), N_3; j, N_2)] \\
& = \sum_{N_3=1}^M [V_i^+(N_1, N_3) h(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3) h(-(i - \frac{1}{2}), N_3; j, N_2)] \\
& + \sum_{n=-(i-\frac{3}{2})}^{i-\frac{3}{2}} \sum_{N_4=1}^M \sum_{N_3=1}^M [V_i^+(N_1, N_3) h(i - \frac{1}{2}, N_3; n, N_4) + V_i^-(N_1, N_3) h(-(i - \frac{1}{2}), N_3; n, N_4)] \\
& \quad \times K(n, N_4; j, N_2) \tag{16}
\end{aligned}$$

where we have defined the potentials

$$V_i^+(N_1, N_2) = -[h(i + \frac{1}{2}, N_1; i - \frac{1}{2}, N_2) - h(i, N_1; i - 1, N_2)] \tag{17}$$

$$V_i^-(N_1, N_2) = -[h(i + \frac{1}{2}, N_1; -(i - \frac{1}{2}), N_2) - h(i, N_1; -(i - 1), N_2)] \tag{18}$$

Note that on a polar raster we have $V_i^+(N_1, N_2) = V_i^-(N_1, N_2 + M/2)$. Since the covariance function $K(i, N_1; j, N_2)$ is assumed to be non-negative definite, equation (4) must have a unique solution. The solutions to (15) and (16) must be identical, so that

$$\Delta h(i, N_1; j, N_2) = \sum_{N_3=1}^M [V_i^+(N_1, N_3) h(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3) h(-(i - \frac{1}{2}), N_3; j, N_2)] \tag{19}$$

Equation (19) is the basic recurrence that is the heart of the Levinson-like algorithm. The left side is the difference of two two-dimensional discrete Laplacian operators, analogous to the difference of one-dimensional discrete Laplacian operators appearing in the split algorithms of [6]. The right side generalizes the three-term recurrence in [6] to a multi-term recurrence; this is analogous to the matrix recurrence in [7]. However, it is applicable to non-symmetric block Toeplitz-plus-Hankel systems, unlike that of [7]. Writing out (19) explicitly, we have

$$\begin{aligned}
h(i + \frac{1}{2}, N_1; j, N_2) &= h(i, N_1; j + \frac{1}{2}, N_2) + h(i, N_1; j - \frac{1}{2}, N_2) - h(i - \frac{1}{2}, N_1; j, N_2) \\
&+ h(i - \frac{1}{2}, N_1; j, N_2 + 1) + h(i - \frac{1}{2}, N_1; j, N_2 - 1) - h(i - \frac{1}{2}, N_1 + 1; j, N_2) - h(i - \frac{1}{2}, N_1 - 1; j, N_2) \\
&+ \sum_{N_3=1}^M [V_i^+(N_1, N_3)h(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3)h(-(i - \frac{1}{2}), N_3; j, N_2)] \quad (20)
\end{aligned}$$

for all $-(i - \frac{3}{2}) \leq j \leq (i - \frac{3}{2})$ and $1 \leq N_1, N_2 \leq M$. Although we have implicitly treated i as positive throughout the derivations, the recursive equations hold for negative i as well. When i is an integer and j is a half-integer, equation (20) will update h from the real points to the interpolated points. When i is a half-integer and j is an integer, equation (20) will update h from the interpolated points to the real points.

III DERIVATION OF THE SCHUR-LIKE ALGORITHM

A. Derivation of the Schur-Like Recurrence

We still need to calculate the potentials $V_i^+(N_1, N_2)$ and $V_i^-(N_1, N_2)$ at the beginning of every update so that we can use the recursive formula (20). To do this, we introduce the *Schur variables* (defined at integer and half-integer points)

$$s(i, N_1; j, N_2) \triangleq \delta_{i, N_1; j, N_2} + K(i, N_1; j, N_2) - h(i, N_1; j, N_2) - \sum_{n=-(i-1)}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) K(n, N_3; j, N_2) \quad (21)$$

where $\delta_{i, N_1; j, N_2} = 0$ unless $i = j$ and $N_1 = N_2$, in which case it is unity.

Since the Schur variables are linear combinations of the prediction *error* filters $\delta_{i,N_1;j,N_2} - h(i, N_1; j, N_2)$, equation (17)-(20) show that $s(i, N_1; j, N_2)$ satisfies the recurrence (20), but now for *all* j :

$$\begin{aligned} s(i + \frac{1}{2}, N_1; j, N_2) &= s(i, N_1; j + \frac{1}{2}, N_2) + s(i, N_1; j - \frac{1}{2}, N_2) - s(i - \frac{1}{2}, N_1; j, N_2) \\ &+ s(i - \frac{1}{2}, N_1; j, N_2 + 1) + s(i - \frac{1}{2}, N_1; j, N_2 - 1) - s(i - \frac{1}{2}, N_1 + 1; j, N_2) - s(i - \frac{1}{2}, N_1 - 1; j, N_2) \\ &+ \sum_{N_3=1}^M [V_i^+(N_1, N_3)s(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3)s(-(i - \frac{1}{2}), N_3; j, N_2)] \end{aligned} \quad (22)$$

Equation (22) is the basic recurrence for the Schur-like algorithm. Note that for $-(i - 1) \leq j \leq (i - 1)$ $s(i, N_1; j, N_2) = 0$ by (4).

B. Computation of Potentials

Setting $j = (i - \frac{1}{2})$ and $-(i - \frac{1}{2})$ in (22), we have

$$\begin{aligned} \sum_{N_3=1}^M [V_i^+(N_1, N_3)s(i - \frac{1}{2}, N_3; i - \frac{1}{2}, N_2) + V_i^-(N_1, N_3)s(-(i - \frac{1}{2}), N_3; i - \frac{1}{2}, N_2)] \\ = s(i - \frac{1}{2}, N_1; i - \frac{1}{2}, N_2) - s(i, N_1; i, N_2) + \Delta_\theta s(i, N_1; i - \frac{1}{2}, N_2) \end{aligned} \quad (23)$$

$$\begin{aligned} \sum_{N_3=1}^M [V_i^+(N_1, N_3)s(i - \frac{1}{2}, N_3; -(i - \frac{1}{2}), N_2) + V_i^-(N_1, N_3)s(-(i - \frac{1}{2}), N_3; -(i - \frac{1}{2}), N_2)] \\ = s(i - \frac{1}{2}, N_1; -(i - \frac{1}{2}), N_2) - s(i, N_1; -i, N_2) + \Delta_\theta s(i, N_1; -(i - \frac{1}{2}), N_2) \end{aligned} \quad (24)$$

Equations (23) and (24) can be written in matrix notation as

$$\hat{\mathbf{V}}^+ \bar{\mathbf{S}}^{++} + \hat{\mathbf{V}}^- \bar{\mathbf{S}}^{-+} = \mathbf{X}^+ \quad (25)$$

$$\hat{\mathbf{V}}^+ \bar{\mathbf{S}}^{+-} + \hat{\mathbf{V}}^- \bar{\mathbf{S}}^{--} = \mathbf{X}^- \quad (26)$$

where we have defined the $M \times M$ matrices

$$[\hat{\mathbf{V}}^+]_{N_1, N_2} = V_i^+(N_1, N_2), [\hat{\mathbf{V}}^-]_{N_1, N_2} = V_i^-(N_1, N_2) \quad (27)$$

$$[\bar{\mathbf{S}}^{\pm\pm}]_{N_1, N_2} = s(\pm(i - \frac{1}{2}), N_1; \pm(i - \frac{1}{2}), N_2) \quad (28)$$

$$[\mathbf{X}^\pm]_{N_1, N_2} = s(i - \frac{1}{2}, N_1; \pm(i - \frac{1}{2}), N_2) - s(i, N_1; i, N_2) + \Delta_\theta s(i, N_1; \pm(i - \frac{1}{2}), N_2) \quad (29)$$

If the system matrix defined in (4) (written explicitly in (44) below) is strongly non-singular, i.e. the leading principal submatrices are all non-singular, then (25) and (26) can be solved in closed form as

$$\bar{\mathbf{V}}^+ = (\mathbf{X}^+ - \mathbf{X}^-(\bar{\mathbf{S}}^{--})^{-1}\bar{\mathbf{S}}^{+-})(\bar{\mathbf{S}}^{++} - \bar{\mathbf{S}}^{+-}(\bar{\mathbf{S}}^{--})^{-1}\bar{\mathbf{S}}^{+-})^{-1} \quad (30)$$

$$\bar{\mathbf{V}}^- = (\mathbf{X}^- - \mathbf{X}^+(\bar{\mathbf{S}}^{++})^{-1}\bar{\mathbf{S}}^{+-})(\bar{\mathbf{S}}^{--} - \bar{\mathbf{S}}^{+-}(\bar{\mathbf{S}}^{++})^{-1}\bar{\mathbf{S}}^{+-})^{-1} \quad (31)$$

The strongly non-singular assumption is necessary and sufficient for (25) and (26) to have a unique solution; the proof of this is a direct generalization of the one in [14]. A similar assumption is required by the standard multichannel Levinson and Schur algorithms.

The split Schur-like algorithm consists of computing $s(i, N_1; j, N_2)$ by propagating (22), initialized using $K(i, N_1 - N_2)$, while computing the $V_i^\pm(N_1, N_2)$ from the $s(i, N_1; j, N_2)$ using (28)-(31).

C. Summary of Overall Procedure

The overall procedure can be summarized as follows. Let I_{max} be the largest radius (maximum radial prediction order). Then:

1. Initialization of Split Schur-Like Algorithm

$$\mathbf{H}_{\pm\frac{1}{2}, 0} = \mathbf{K}_{\pm\frac{1}{2}, 0}(\mathbf{I} + \mathbf{K}_{0,0})^{-1}, \quad \mathbf{H}_{\pm 1, 0} = \mathbf{K}_{\pm 1, 0}(\mathbf{I} + \mathbf{K}_{0,0})^{-1}$$

where $[\mathbf{K}_{\pm\frac{1}{2}, 0}]_{N_1, N_2} = K(\pm\frac{1}{2}, N_1; 0, N_2)$, $[\mathbf{K}_{0,0}]_{N_1, N_2} = K(0, N_1; 0, N_2)$,

$$[\mathbf{K}_{\pm 1, 0}]_{N_1, N_2} = K(\pm 1, N_1; 0, N_2)$$

$$s(\pm\frac{1}{2}, N_1; j, N_2) = \delta_{\pm\frac{1}{2}, N_1; j, N_2} + K(\pm\frac{1}{2}, N_1; j, N_2) - \sum_{N_3=1}^M h(\pm\frac{1}{2}, N_1; 0, N_3)K(0, N_3; j, N_2)$$

$$\text{for all } j = \pm\frac{1}{2}, \dots, \pm 2I_{max} \text{ and } N_1, N_2 = 1, \dots, M$$

$$s(\pm 1, N_1; j, N_2) = \delta_{\pm 1, N_1; j, N_2} + K(\pm 1, N_1; j, N_2) - h(\pm 1, N_1; j, N_2) - \sum_{N_3=1}^M h(\pm 1, N_1; 0, N_3)K(0, N_3; j, N_2)$$

$$\text{for all } j = \pm 1, \dots, \pm 2I_{max} \text{ and } N_1, N_2 = 1, \dots, M$$

2. Propagation of Split Schur-Like Algorithm

A. Computation of the potentials $V_i^+(N_1, N_2)$ and $V_i^-(N_1, N_2)$:

Compute $V_i^+(N_1, N_2)$ and $V_i^-(N_1, N_2)$ from the available $s(\pm(i-\frac{1}{2}), N_1; \pm(i-\frac{1}{2}), N_2)$ and $s(\pm i, N_1; \pm i, N_2)$ using equations (30) and (31);

B. Update the Schur variables

For $j = \pm(i + \frac{1}{2})$ To $j = \pm 2I_{max}$, $N_1 = 1$ To M , $N_2 = 1$ To M , Parallel Do

Update the Schur variables using (22).

End Parallel Do $\{j, N_1, N_2\}$.

3. Propagation of Split Levinson-Like Recurrence

A. Propagate the Boundary Points

For $N_1 = 1$ To M , $N_2 = 1$ To M , Parallel Do

$$h(i + \frac{1}{2}, N_1; i - \frac{1}{2}, N_2) = h(i, N_1; i - 1, N_2) - V_i^+(N_1, N_2) \quad (32)$$

$$h(i + \frac{1}{2}, N_1; -(i - \frac{1}{2}), N_2) = h(i, N_1; -(i - 1), N_2) - V_i^-(N_1, N_2) \quad (33)$$

End Parallel Do $\{N_1, N_2\}$.

B. Propagate Non-Boundary Points

For $j = -(i - \frac{1}{2})$ To $j = (i - \frac{1}{2})$, $N_1 = 1$ To M , $N_2 = 1$ To M , Parallel Do

Update $h(i, N_1; j, N_2)$ using equation (20).

End Parallel Do $\{j, N_1, N_2\}$.

4. Repeat steps 2 and 3 from $i = 1$ to I_{max} with increment $\frac{1}{2}$.

Note that the Levinson and Schur recurrences (20) and (22) have identical forms, with complementary supports. Hence they can be propagated in parallel using identical processors; this possibility was first noted for the one-dimensional case in [16].

IV RANDOM FIELDS WITH BLOCK TOEPLITZ-PLUS-HANKEL COVARIANCES

In the above derivation, we have assumed that the covariance function is already known. If only a sequence of two-dimensional time series data are available, there are two methods for obtaining a covariance function having the desired Toeplitz-plus-Hankel structure (13),(14). The first method is to compute a data covariance matrix, and then determine a symmetric Toeplitz-plus-Hankel block Toeplitz-plus-Hankel matrix close (in some sense) to this matrix. This is a two-dimensional Toeplitz-plus-Hankel generalization of the well known "Toeplitzation" problem [17]. Some procedures for this problem are suggested in [18]. The second method is to assume that the data are generated by some underlying model, for which unknown parameters may need to be determined.

In this section we focus on the second approach, giving some specific examples of random fields whose covariances satisfy assumptions (13) and (14). These are merely illustrative; there are of course many others. We also note how the algorithms of Section II relate to the continuous-parameter algorithms of [13].

A. Isotropic Random Fields

For an isotropic random field, the covariance is a function of distance only, i.e., if x and y are two arbitrary points in the plane, then $K(x, y) = K(|x - y|)$. Consider the special case of a isotropic random field with covariance $K(x, y) = \rho^{|x-y|^2}$, which is often used in image modeling [19]. In polar coordinates on a discrete polar raster, this covariance function can be represented as

$$\begin{aligned} K(i, N_1; j, N_2) &= \rho^{i^2+j^2-2ij\cos(2\pi(N_1-N_2)/M)} \\ &= \rho^{\frac{1}{2}[(i+j)^2+(i-j)^2]-[(i+j)^2-(i-j)^2]\cos(2\pi(N_1-N_2)/M)} \\ &\approx 1 + \frac{1}{2}([(i+j)^2+(i-j)^2] - [(i+j)^2-(i-j)^2]\cos(2\pi(N_1-N_2)/N)) \ln \rho \end{aligned} \quad (34)$$

if $\rho \approx 1$

Note that the exponent has the Toeplitz-plus-Hankel structure required by (13) and (14), and that it is not merely Toeplitz in i and j ; hence the multichannel Levinson algorithm is not applicable. If $\rho \approx 1$, the entire covariance satisfies (13) and (14). Indeed, *any* slowly-changing function of distance on a polar raster satisfies (13) and (14) in its radial and angular arguments.

B. Separable Covariance Functions

A separable covariance function is one that can be decomposed into multiplication of a function of the radial part and a function of the transverse part, i.e., the covariance function $K(i, N_1; j, N_2)$ can be expressed as

$$K(i, N_1; j, N_2) = R(i, j) \times T(N_1, N_2) \quad (35)$$

for some functions R and T . This type of covariance function satisfies (13) and (14) as long as both R and T have Toeplitz-plus-Hankel structure. Examples include:

1. 2-D Discrete Wiener Process

The 2-D discrete Wiener process on a polar raster can be defined as

$$x_{i, N_1} = \sum_{j=0}^i \sum_{n=1}^M w_{j, n} \quad , \quad x_{0, N_1} = 0 \quad (36)$$

where $w_{j, n}$ is a zero-mean discrete white noise field with variance σ^2 . Its covariance function is equal to

$$\begin{aligned} K(i, N_1; j, N_2) &= E[x_{i, N_1} x_{j, N_2}] = M\sigma^2 \min(i, j) \\ &= M\sigma^2 \frac{1}{2} [|i + j| - |i - j|] \end{aligned} \quad (37)$$

Note that $R(i, j)$ has Toeplitz-plus-Hankel structure and $T(N_1, N_2)$ is a constant function.

2. 2-D Circularly Symmetric Markovian Random Field

In a first-order 2-D circularly symmetric Markovian random field, the output is a uniformly linear combination of the previous "shell" of data plus white noise, i.e.

$$x_{i, N_1} = a \sum_{n=1}^M x_{i-1, n} + w_{i, N_1} \quad (38)$$

If $x_{0,n}$ is assumed to be zero for all n , and the variance of w_{i,N_1} is equal to σ^2 , then the covariance function is

$$\begin{aligned} K(i, N_1; j, N_2) &= E[x_{i,N_1}; x_{j,N_2}] \\ &= \frac{M\sigma^2}{1-a^2} [a^{|i-j|} - a^{|i+j|}] \end{aligned} \quad (39)$$

Again, $R(i, j)$ has Toeplitz-plus-Hankel structure and $T(N_1, N_2)$ is a constant function. In the limit $a \rightarrow 0$ (39) reduces to (37).

C. Relations with Continuous Algorithms

It is instructive to examine the continuous-parameter limits of some of the equations of this paper. Let the intervals between points be δ_r in the radial direction and $\delta_\theta = \frac{2\pi}{M}$ radians in the transverse (angular) direction. Introducing a radial weighting factor, as discussed earlier, and taking limits as δ_r and δ_θ go to zero result in the following transformations:

1. The discretized Wiener-Hopf equation (6) becomes the Wiener-Hopf integral equation (5);
2. $\delta_{i,N_1;j,N_2}$ becomes a continuous two-dimensional impulse function, dominating the other terms in the definition (21) of the Schur variables, so that (30) and (31) may be replaced with $\bar{V}^+ \approx X^+$ and $\bar{V}^- \approx X^-$. Using this, equation (29) becomes

$$V(x, \theta_1; \theta_2) = -\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y}\right)s(x, \theta_1; y = x, \theta_2) \quad (40)$$

where x and y are continuous radii and θ_1 and θ_2 are continuous angles. Equation (40) has the form of (4-17b) of [13]. Similarly, the continuous version of (13) has the form of (4-2) of [13];

3. Equation (15), with its difference of discrete two-dimensional Laplacian operators on the left side, is clearly analogous to $(\Delta_x \triangleq \text{Laplacian with respect to } x)$

$$(\Delta_x - \Delta_y)h(x, \theta_1; y, \theta_2) = \int_0^{2\pi} V(x, \theta_1; \theta_3)h(x, \theta_3; y, \theta_2) d\theta_3 \quad (41)$$

which is the two-dimensional form of (4-1) of [13]. However, (41) is NOT the continuous limit of (15) with radial weighting, since $\frac{1}{\sqrt{x}} \frac{d^2}{dx^2} (\sqrt{x} f(x)) = (\frac{d^2}{dx^2} + \frac{1}{x} \frac{d}{dx} - \frac{1}{4x^2}) f(x)$, which is not the radial part of the 2-D Laplacian. On the other hand, $\frac{1}{x} \frac{d^2}{dx^2} (x f(x)) = (\frac{d^2}{dx^2} + \frac{2}{x} \frac{d}{dx}) f(x)$, which is the radial part of the 3-D Laplacian. This shows that the results of [13], derived for the continuous 3-D case, do not apply *exactly* to the 2-D case (as do the results of this paper);

4. The algorithms of this paper require the differences of the radial parts and transverse parts of the Laplacian of the covariance to be *separately* zero: (13) and (14) must be *separately* zero. However, in the continuous limit, we have $h(i, N_1; n, N_3) \approx h(i - \frac{1}{2}, N_1; n, N_3)$, and the last two sums in (11) may be combined. Then it suffices for the *sum* $(\Delta_r + \Delta_\theta)K(i, N_1; j, N_3) = 0$, rather than (13) and (14) separately. This agrees with the requirement $(\Delta_x - \Delta_y)K(x, y) = 0$ for the algorithms in [13].

D. Application to Discretized Continuous-Parameter Problems

We can draw some important conclusions from these observations. If the algorithms of this paper are being used to solve the discretized version (6) of the Wiener-Hopf equation (5), then :

1. Equations (30) and (31) may be replaced with the approximations $\bar{V}^+ \approx X^+$ and $\bar{V}^- \approx X^-$;
2. By the chain rule, *any* continuous function of the distance between two points will satisfy (13) and (14), since the square of the distance itself does. Hence the algorithms may be used for any isotropic random field. Note in particular that (32) becomes

$$K(i, N_1; j, N_2) = \rho^{\delta_r^2(i^2 + j^2 - 2ij \cos(N_1 - N_2)\delta_\theta)} \quad (42)$$

and $\rho^{\delta_r^2} \rightarrow 1$ as $\delta_r \rightarrow 0$;

3. Conditions (13) and (14) may be replaced with the more general condition

$$(\Delta_r + \Delta_\theta)K(i, N_1; j, N_2) = 0.$$

Numerical studies have shown that approximation (4) gives very good results for $\delta_r \approx 0.001$, but approximation (6) is much more sensitive to non-infinitesimal δ_r .

V COMPLEXITY AND GENERAL TOEPLITZ-PLUS-HANKEL SYSTEMS

A. Computational Complexity

We determine the number of Multiplications-And-Divisions (MADs) needed to solve (4) up to order $i = I_{max}$. Although some current DSP chips can perform multiplications as quickly as additions, the fact remains that multiplication is a more complex operation than addition. Also, the computational savings in the number of additions is similar to that for MADs, although we omit details.

The initialization of the Levinson-like recurrences requires $2 M \times M$ matrix inversions and $4 M \times M$ matrix multiplications, or $2(\frac{M^3}{3} + \frac{M^2}{2}) + 4M^3$ MADs. The initialization of the Schur-like recurrences requires $8I_{max} M \times M$ matrix multiplications, or $8I_{max}M^3$ MADs. Each Schur-like recursion update of $s(i, N_1; j, N_2)$ from i to $i + \frac{1}{2}$ requires $16(I_{max} - i)M^2$ MADs. Computation of the potentials requires $4 M \times M$ matrix inversions and $6 M \times M$ matrix multiplications. Finally, updating $h(i, N_1; j, N_2)$ from i to $i + \frac{1}{2}$ in the Levinson-like recurrence requires $4(2i + 1)M^2$ MADs. The total number of multiplications needed to solve (4) up to $i = I_{max}$ is

$$\begin{aligned} 4M^3 + 2(\frac{M^3}{3} + \frac{M^2}{2}) + 8I_{max}M^3 + 2 \sum_{i=1}^{I_{max}} [16(I_{max} - i)M^2 + (4(\frac{M^3}{3} + \frac{M^2}{2}) + 6M^3) + 4(2i + 1)M^2] \\ = 24I_{max}^2M^2 + I_{max}(\frac{68M^3}{3} + 4M^2) + 4M^3 + 2(\frac{M^3}{3} + \frac{M^2}{2}) \end{aligned} \quad (43)$$

This can be seen to be $O(I_{max}^2M^2)$ MADs if $I_{max} \gg M \gg 1$. Solution of (4) using Gaussian elimination would require $\frac{(2I_{max}M)^3}{3} + \frac{(2I_{max}M)^2}{2} = O(I_{max}^3M^3)$ MADs. Hence the savings in MADs over Gaussian elimination for large I_{max} and M is a factor of order $I_{max}M$.

B. Comparison with Reformulation as a Block-Toeplitz System

In [20] Merchant and Parks noted that a Toeplitz-plus-Hankel system of equations can be reformulated as a block-Toeplitz system of equations with 2×2 blocks. Although no multichannel generalizations were discussed in [20], it is not difficult to show that a system of equations in which the system matrix is

the sum of a block-Toeplitz matrix and a block-Hankel matrix, where the blocks are $M \times M$, can be reformulated as a block-Toeplitz system of equations with $2M \times 2M$ blocks. This could then be solved using the multichannel Levinson algorithm. We now compare this approach, which we call the generalized Merchant-Parks procedure, to the algorithm of this paper.

If the generalized Merchant-Parks procedure is used to solve (4) up to order $i = I_{max}$, the number of MADs required is $32I_{max}^2 M^3 + I_{max}(\frac{8M^3}{3} + 2M^2)$, since $2M \times 2M$ matrices are being multiplied and propagated. Hence if $I_{max} \gg M \gg 1$ the algorithm of this paper requires roughly $\frac{3}{4M}$ as many MADs as the generalized Merchant-Parks procedure; for large M this can be quite significant. If $M = 1$ the algorithm of this paper reduce to that of [14], which requires roughly 75% as many MADs as the original Merchant-Parks procedure [20].

On the other hand, the algorithm of this paper requires that the system matrix be block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks, while the generalized Merchant-Parks algorithm does not require the blocks to have special structure. Thus the generalized Merchant-Parks algorithm requires more computation, but solves a more general problem.

C. Solution of Arbitrary Toeplitz-plus-Hankel Block Toeplitz-plus-Hankel Systems

Equation (4) can be written as the following Toeplitz-plus-Hankel block Toeplitz-plus-Hankel system:

$$\begin{bmatrix} 0 & \mathbf{H}_{i, -(i-1)} & \cdots & \mathbf{H}_{i, (i-1)} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{H}_{-i, -(i-1)} & \cdots & \mathbf{H}_{-i, (i-1)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} + \mathbf{K}_{-i, -i} & \cdots & \mathbf{K}_{-i, i} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{i, -i} & \cdots & \mathbf{I} + \mathbf{K}_{i, i} \end{bmatrix} = - \begin{bmatrix} \tilde{\mathbf{S}}_{i, -i} & 0 & \cdots & 0 & \tilde{\mathbf{S}}_{i, i} \\ \tilde{\mathbf{S}}_{-i, -i} & 0 & \cdots & 0 & \tilde{\mathbf{S}}_{-i, i} \end{bmatrix} \quad (44)$$

where

$$[\mathbf{H}_{\pm i, \pm j}]_{N_1, N_2} \triangleq h(\pm i, N_1; \pm j, N_2), \quad j = -(i-1), \dots, (i-1), 1 \leq N_1, N_2 \leq M \quad (45)$$

$$[\mathbf{K}_{j, l}]_{N_1, N_2} \triangleq K(j, N_1; l, N_2), \quad j, l = -(i-1), \dots, (i-1), 1 \leq N_1, N_2 \leq M \quad (46)$$

$$[\tilde{S}_{\pm i, \pm i}]_{N_1, N_2} \triangleq s(\pm i, N_1; \pm i, N_2), 1 \leq N_1, N_2 \leq M \quad (47)$$

In (44)-(47) \mathbf{I} is the $M \times M$ identity matrix and $\mathbf{0}$ is an $M \times M$ matrix of zeros. Conditions (13) and (14) are equivalent to requiring that the system matrix in (44) be block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks.

In this section we solve a Toeplitz-plus-Hankel block Toeplitz-plus-Hankel system of equations having the same system matrix as (44), but with an arbitrary right side. This system is

$$\begin{bmatrix} \mathbf{X}_{-i} & \mathbf{X}_{-(i-1)} & \cdots & \mathbf{X}_{i-1} & \mathbf{X}_i \end{bmatrix} \begin{bmatrix} \mathbf{I} + \mathbf{K}_{-i, -i} & \cdots & \mathbf{K}_{-i, i} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{i, -i} & \cdots & \mathbf{I} + \mathbf{K}_{i, i} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{-i} & \mathbf{B}_{-(i-1)} & \cdots & \mathbf{B}_{i-1} & \mathbf{B}_i \end{bmatrix} \quad (48)$$

where the right side is arbitrary. Recall that the algorithms of this paper do not require the system matrix to be symmetric. To find the solution $\bar{\mathbf{X}} (\triangleq [\mathbf{X}_{-i}, \dots, \mathbf{X}_i])$, note that from the definition (21) of $s(i, N_1; j, N_2)$ we have

$$\begin{bmatrix} \bar{\mathbf{H}}_m \\ \bar{\mathbf{H}}_{-m} \end{bmatrix} \begin{bmatrix} \mathbf{I} + \mathbf{K}_{-i, -i} & \cdots & \mathbf{K}_{-i, i} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{i, -i} & \cdots & \mathbf{I} + \mathbf{K}_{i, i} \end{bmatrix} = - \begin{bmatrix} \tilde{S}_{i-m+1, -i} & \cdots & \tilde{S}_{i-m+1, -(i-m+1)} & \mathbf{0} & \cdots & \mathbf{0} & \tilde{S}_{i-m+1, i-m+1} & \cdots & \tilde{S}_{i-m+1, i} \\ \tilde{S}_{-(i-m+1), -i} & \cdots & \tilde{S}_{-(i-m+1), -(i-m+1)} & \mathbf{0} & \cdots & \mathbf{0} & \tilde{S}_{-(i-m+1), i-m+1} & \cdots & \tilde{S}_{-(i-m+1), i} \end{bmatrix} \quad (49)$$

where

$$\bar{\mathbf{H}}_m \equiv [\underbrace{\mathbf{0}, \dots, \mathbf{0}}_m, \mathbf{H}_{i-m+1, -(i-m)}, \dots, \mathbf{H}_{i-m+1, i-m}, -\mathbf{I}, \underbrace{\mathbf{0}, \dots, \mathbf{0}}_{m-1}] \quad (50)$$

$$\bar{\mathbf{H}}_{-m} \equiv [-\mathbf{I}, \underbrace{\mathbf{0}, \dots, \mathbf{0}}_{m-1}, \mathbf{H}_{-(i-m+1), -(i-m)}, \dots, \mathbf{H}_{-(i-m+1), i-m}, \underbrace{\mathbf{0}, \dots, \mathbf{0}}_m] \quad (51)$$

Equation (44) is a special case of (49) with $m = 1$.

Assume that all of the central submatrices of the system (44) are non-singular. Then the unique solution to (48) can be expressed as a linear combination of $\tilde{\mathbf{H}}_m, m = -i, \dots, i$ by using (49)

$$\bar{\mathbf{X}} = \sum_{m=-i, m \neq 0}^i \mathbf{C}_m \tilde{\mathbf{H}}_m \quad (\text{i.e. } \mathbf{X}_j = \sum_{l=-i}^i \mathbf{C}_l \mathbf{H}_{l,j}). \quad (52)$$

Here \mathbf{C}_m can be found by equating the linear combination (52) to (48), for $1 \leq j \leq (i-1)$:

$$\mathbf{C}_{-j} \tilde{\mathbf{S}}_{-j,-j} + \mathbf{C}_j \tilde{\mathbf{S}}_{j,-j} = -(\mathbf{B}_{-j} + \sum_{n=-(j-1)}^{j-1} \mathbf{C}_n \tilde{\mathbf{S}}_{n,-j}) \quad (53)$$

$$\mathbf{C}_{-j} \tilde{\mathbf{S}}_{-j,j} + \mathbf{C}_j \tilde{\mathbf{S}}_{j,j} = -(\mathbf{B}_j + \sum_{n=-(j-1)}^{j-1} \mathbf{C}_n \tilde{\mathbf{S}}_{n,j}). \quad (54)$$

The overall procedure is as follows. Compute the $\mathbf{H}_{i,j}$ and $\tilde{\mathbf{S}}_{i,j}$ using the Levinson-like and Schur-like algorithms. Next, recursively compute $\mathbf{C}_{\pm j}$ in increasing j by solving the $2M \times 2M$ systems (53) and (54). Finally, compute $\bar{\mathbf{X}}$ using (52).

The procedures in (52-54) require roughly $4I_{max}^2 M^3$ MADs, which for $I_{max} \gg M \gg 1$ dominates the $24I_{max}^2 M^2$ MADs that is the dominant term in the number of MADs (43) required by the basic algorithm. For an arbitrary right side, the generalized Merchant-Parks algorithm requires $48I_{max}^2 M^3$ MADs. Thus the algorithm of this section requires only $\frac{1}{12}$ as many MADs when $I_{max} \gg M \gg 1$.

VI CONCLUSION

New fast algorithms for solving the discrete two-dimensional Wiener-Hopf equation on a polar raster when the covariance function has Toeplitz-plus-Hankel structure have been derived. Since we have performed explicitly discrete derivations, instead of just discretizing the continuous versions [13], the algorithms do not require fine discretization or closely-spaced points; if adjacent points are close enough, then the algorithms reduce to the continuous case [13]. In particular, the proposed fast algorithms make full use of the Toeplitz-plus-Hankel structure of the covariance function, so that the overall computational complexity is only $O(I_{max}^2 M^2)$ MADs, as opposed to $O(I_{max}^2 M^3)$ MADs for the generalized Merchant-Parks algorithm discussed in the paper and $O(I_{max}^3 M^3)$ MADs for Gaussian elimination. These algorithms are also highly parallelizable, making them even more favorable in a vector/parallel processor environment.

The *smoothing* filter for estimating the points *inside* the disk of observations can be computed from the prediction filters using a generalized discrete Bellman-Siebert-Krein identity, as was done for the one-dimensional continuous case in [9]. The overall complexity is reduced compared with Gaussian elimination. This is considered in the separate paper [8].

Unresolved issues include mapping of this algorithm into optimal array processor architectures, the numerical stability of the algorithm, and practical applications of this algorithm in problems such as image restoration and coding. Preliminary results on these issues have been encouraging.

ACKNOWLEDGMENT

The work of both authors was supported by the Air Force Office of Scientific Research under grant # AFOSR-89-0017.

References

- [1] D.E. Dudgeon and R.E. Mersereau, *Multi-Dimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [2] J.H. Justice, *A Levinson-type algorithm for two-dimensional Wiener filtering using bivariate Szego polynomials*, Proc. IEEE, vol. 65, no.6, pp. 882-886, June 1977.
- [3] T. Marzetta, *A linear prediction approach to two-dimensional spectral factorization and spectral estimation*, Ph.D. Thesis, Dept. of EECS, M.I.T., Cambridge, MA, Feb. 1978.
- [4] G.T. Herman ed., *Image Reconstruction from Projections: Implementation and Applications*, Springer Verlag, NY, 1979.
- [5] J.L. Walker, *Range-Doppler imaging of rotating objects*, IEEE Trans. Aerosp. Electron. Syst. vol. AES-16, pp. 23-52, Jan. 1980.
- [6] P. Delsarte and Y. Genin, *On the splitting of classical algorithms in linear prediction theory*, IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-35, no.5, pp. 645-653, May 1987.

- [7] P. Delsarte and Y. Genin, *Multichannel singular predictor polynomials*, IEEE Trans. Circuit and Systems, vol. CAS-35, no.2, pp. 190-200, Feb. 1988.
- [8] W.-H. Fang and A.E. Yagle, *Fast algorithms for linear least-squares smoothing problems in one and two Dimensions Using generalized discrete Bellman-Siebert-Krein resolvent identities*, submitted to IEEE Trans. ASSP, Oct. 1989.
- [9] T. Kailath, *Application of a resolvent identity to a linear smoothing problem*, SIAM J. Cont., vol. 7, no. 1, pp. 68-74, Feb. 1969.
- [10] B.C. Levy and J.N.Tsitsiklis, *A fast algorithm for linear estimation of two-dimensional isotropic random fields*, IEEE Trans. on Info. Th., vol. IT-31, no.5, pp. 635-644, Sept. 1985.
- [11] A.E. Yagle, *A fast algorithm for linear estimation of three-dimensional homogeneous anisotropic random fields*, Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc., Dallas, TX, April 1987, pp. 1712-1715.
- [12] A.E. Yagle, *Generalized Levinson and fast Cholesky algorithms for three-dimensional random fields estimation problem*, Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc., New York, April 1988, pp. 1060-1063.
- [13] A.E. Yagle, *Analogues of split Levinson, Schur, and lattice algorithms for three-dimensional random field estimation problem*, to appear in SIAM J. Appl. Math.
- [14] A.E. Yagle, *New analogues of split algorithms for arbitrary Toeplitz-plus-Hankel matrices*, to appear in IEEE Trans. ASSP.
- [15] W.-H. Fang, A.E. Yagle, *Discrete fast algorithms for two-dimensional linear prediction on a polar raster*, Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc., Albuquerque, NM, April 1990, pp. 2017-2020.
- [16] S.Y. Kung and Y.H. Hu, *A highly concurrent algorithm and pipelined architecture for solving Toeplitz systems*, IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-31, no.1, pp. 66-75, Feb. 1983.

- [17] H. Wang, G. Wakefield, *Signal-Subspace Approximation for Line Spectrum Estimation*, Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc., Dallas, TX, April 1987, pp. 2054-2057.
- [18] W.-H. Fang and A.E. Yagle, *Two methods for Toeplitz-plus-Hankel approximation to a data covariance matrix*, submitted to IEEE Trans. ASSP, July 1990.
- [19] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [20] G.A. Merchant and T.W. Parks, *Efficient solution of a Toeplitz-plus-Hankel coefficient matrix system of equations*, IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-30, no.1, pp. 40-44, Feb. 1982.

FIGURE HEADING

1. The polar raster on which the two-dimensional random field is defined, where $M = 8$

APPENDIX D

A. Basic Problem

The problem considered is as follows. From noisy observations $\{y_{i,N,M}\}$ of a zero-mean real-valued discrete random field $\{x_{i,N,M}\}$ at the points (i, N, M) inside a sphere, compute the linear least-squares estimate of $x_{i,N,M}$ for all points on the edge of the sphere. Here i is an integer radius from the origin, and N and M are the integer indices of the arguments (angles).

The observations $\{y_{i,N,M}\}$ are related to the field $x_{i,N,M}$ by $y_{i,N,M} = x_{i,N,M} + v_{i,N,M}$, where $\{v_{i,N,M}\}$ is a zero-mean discrete white noise field with unit power, and $\{x_{i,N,M}\}$ and $\{v_{i,N,M}\}$ are uncorrelated. The covariance of $\{x_{i,N,M}\}$, $E[x_{i,N_1,M_1}x_{j,N_2,M_2}] = K(i, N_1, M_1; j, N_2, M_2)$, is assumed to be a non-negative definite function with Toeplitz-plus-Hankel structure shown in (6) and (7). The estimates of $x_{i,N,M}$ at the edge of the sphere are computed from the observations $\{y_{i,N,M}\}$ using

$$\hat{x}_{i,N_1,M_1} = \sum_{j=0}^{i-1} \sum_{N_2=1}^N \sum_{M_2=1}^M h(i, N_1, M_1; j, N_2, M_2) y_{j,N_2,M_2} \quad (1)$$

The optimal prediction filters $h(i, N_1, M_1; j, N_2, M_2)$ are computed by solving the three-dimensional discrete Wiener-Hopf equation

$$\begin{aligned} K(i, N_1, M_1; j, N_2, M_2) &= h(i, N_1, M_1; j, N_2, M_2) \\ &+ \sum_{n=-(i-1)}^{i-1} \sum_{N_3=1}^N \sum_{M_3=1}^M h(i, N_1, M_1; n, N_3, M_3) K(n, N_3, M_3; j, N_2, M_2) \end{aligned} \quad (2)$$

for all $-(i-1) \leq j \leq i-1$, $1 \leq N_1, N_2 \leq N$ and $1 \leq M_1, M_2 \leq M$. The goal is to derive a fast algorithm for solving (2) when $K(i, N_1, M_1; j, N_2, M_2)$ has the Toeplitz-plus-Hankel structure shown in (6) and (7) below.

B. Derivation of the Levinson-Like Recurrence

Define the discrete wave operators Δ_r , Δ_θ and Δ_ϕ by

$$\Delta_r f(i, N_1, M_1; j, N_2, M_2) = f(i + \frac{1}{2}, N_1, M_1; j, N_2, M_2) + f(i - \frac{1}{2}, N_1, M_1; j, N_2, M_2)$$

$$-f(i, N_1, M_1; j + \frac{1}{2}, N_2, M_2) - f(i, N_1, M_1; j - \frac{1}{2}, N_2, M_2) \quad (3)$$

$$\begin{aligned} \Delta_\theta f(i, N_1, M_1; j, N_2, M_2) = & f(i - \frac{1}{2}, ((N_1 + 1))_1, M_1; j, N_2, M_2) + f(i - \frac{1}{2}, ((N_1 - 1))_1, M_1; j, N_2) \\ & - f(i - \frac{1}{2}, N_1, M_1; j, ((N_2 + 1))_1, M_2) - f(i - \frac{1}{2}, N_1, M_1; j, ((N_2 - 1))_1, M_2) \end{aligned} \quad (4)$$

$$\begin{aligned} \Delta_\phi f(i, N_1, M_1; j, N_2, M_2) = & f(i - \frac{1}{2}, N_1, ((M_1 + 1))_2; j, N_2, M_2) + f(i - \frac{1}{2}, N_1, ((M_1 - 1))_2; j, N_2) \\ & - f(i - \frac{1}{2}, N_1, M_1; j, N_2, ((M_2 + 1))_2) - f(i - \frac{1}{2}, N_1, M_1; j, N_2, ((M_2 - 1))_2) \end{aligned} \quad (5)$$

where Δ_r , Δ_θ and Δ_ϕ can be regarded as discrete versions of the continuous operators $(\frac{\partial^2}{\partial^2 r_1} - \frac{\partial^2}{\partial^2 r_2})$, $(\frac{\partial^2}{\partial^2 \theta_1} - \frac{\partial^2}{\partial^2 \theta_2})$, and $(\frac{\partial^2}{\partial^2 \phi_1} - \frac{\partial^2}{\partial^2 \phi_2})$ for the radial part and transverse parts, respectively, and $((\cdot))_{1(2)}$ means the mod $N(M)$ operation. To save space, we will omit the $((\cdot))$ in the following derivations. We assume that the covariance function has the Toeplitz-plus-Hankel structure that satisfies the following forms

$$\Delta_r K(i, N_1, M_1; j, N_2, M_2) = 0 \quad (6)$$

$$(\Delta_\theta + \Delta_\phi) K(i, N_1, M_1; j, N_2, M_2) = 0 \quad (7)$$

Applying the Laplacian operator $\Delta = \Delta_r + \Delta_\theta + \Delta_\phi$ to the equation (2), we have after some algebra

$$\begin{aligned} h(i + \frac{1}{2}, N_1, M_1; j, N_2, M_2) = & h(i, N_1, M_1; j + \frac{1}{2}, N_2, M_2) + h(i, N_1, M_1; j - \frac{1}{2}, N_2, M_2) \\ & - h(i - \frac{1}{2}, N_1, M_1; j, N_2, M_2) + h(i - \frac{1}{2}, N_1, M_1; j, N_2 + 1, M_2) + h(i - \frac{1}{2}, N_1, M_1; j, N_2 - 1, M_2) \\ & - h(i - \frac{1}{2}, N_1 + 1, M_1; j, N_2, M_2) - h(i - \frac{1}{2}, N_1 - 1, M_1; j, N_2, M_2) + h(i - \frac{1}{2}, N_1, M_1; j, N_2, M_2 + 1) \\ & + h(i - \frac{1}{2}, N_1, M_1; j, N_2 - 1, M_2 - 1) - h(i - \frac{1}{2}, N_1, M_1 + 1; j, N_2, M_2) - h(i - \frac{1}{2}, N_1, M_1 - 1; j, N_2, M_2) \\ & + \sum_{N_3=1}^N \sum_{M_3=1}^M [V_i^+(N_1, M_1; N_3, M_3) h(i - \frac{1}{2}, N_3, M_3; j, N_2, M_2) + V_i^-(N_1, M_1; N_3, M_3) h(-i + \frac{1}{2}, N_3, M_3; j, N_2, M_2)] \end{aligned} \quad (8)$$

for all $-(i - \frac{3}{2}) \leq j \leq (i - \frac{3}{2})$, $1 \leq N_1, N_2 \leq N$ and $1 \leq M_1, M_2 \leq M$. Here we have defined the potentials

$$V_i^+(N_1, M_1; N_2, M_2) = -[h(i + \frac{1}{2}, N_1, M_1; i - \frac{1}{2}, N_2, M_2) - h(i, N_1, M_1; i - 1, N_2, M_2)] \quad (9)$$

$$V_i^-(N_1, M_1; N_2, M_2) = -[h(i + \frac{1}{2}, N_1, M_1; -i + \frac{1}{2}, N_2, M_2) - h(i, N_1, M_1; -i + 1, N_2, M_2)] \quad (10)$$

C. Derivation of the Schur-Like Recurrence

We still need to calculate the potentials $V_i^+(N_1, M_1; N_2, M_2)$ and $V_i^-(N_1, M_1; N_2, M_2)$ at the beginning of every update so that we can use the recursive formula (8). Since an inner product is a bottle neck in a parallel processing environment, we overcome this difficulty by introducing the *Schur variables*

$$\begin{aligned} s(i, N_1, M_1; j, N_2, M_2) &\equiv \delta_{i, N_1, M_1; j, N_2, M_2} + K(i, N_1, M_1; j, N_2, M_2) - h(i, N_1, M_1; j, N_2, M_2) \\ &- \sum_{n=-(i-1)}^{i-1} \sum_{N_3=1}^N \sum_{M_3=1}^M h(i, N_1, M_1; n, N_3, M_3) K(n, N_3, M_3; j, N_2, M_2) \end{aligned} \quad (11)$$

where $\delta_{i, N_1, M_1; j, N_2, M_2} = 0$ unless $i = j$, $N_1 = N_2$ and $M_1 = M_2$, in which case it is unity.

Since the Schur variables are the linear combinations of the prediction error filters $\delta_{i, N_1, M_1; j, N_2, M_2} - h(i, N_1, M_1; j, N_2, M_2)$, equations (8)-(11) show that $s(i, N_1, M_1; j, N_2, M_2)$ satisfies the recurrence (8), but now for *all* j :

$$\begin{aligned} s(i + \frac{1}{2}, N_1, M_1; j, N_2, M_2) &= s(i, N_1, M_1; j + \frac{1}{2}, N_2, M_2) + s(i, N_1, M_1; j - \frac{1}{2}, N_2, M_2) \\ &- s(i - \frac{1}{2}, N_1, M_1; j, N_2, M_2) + s(i - \frac{1}{2}, N_1, M_1; j, N_2 + 1, M_2) + s(i - \frac{1}{2}, N_1, M_1; j, N_2 - 1, M_2) \\ &- s(i - \frac{1}{2}, N_1 + 1, M_1; j, N_2, M_2) - s(i - \frac{1}{2}, N_1 - 1, M_1; j, N_2, M_2) + s(i - \frac{1}{2}, N_1, M_1; j, N_2, M_2 + 1) \\ &+ s(i - \frac{1}{2}, N_1, M_1; j, N_2 - 1, M_2 - 1) - s(i - \frac{1}{2}, N_1, M_1 + 1; j, N_2, M_2) - s(i - \frac{1}{2}, N_1, M_1 - 1; j, N_2, M_2) \\ &+ \sum_{N_3=1}^N \sum_{M_3=1}^M [V_i^+(N_1, M_1; N_3, M_3) s(i - \frac{1}{2}, N_3, M_3; j, N_2, M_2) + V_i^-(N_1, M_1; N_3, M_3) s(-i + \frac{1}{2}, N_3, M_3; j, N_2, M_2)] \end{aligned} \quad (12)$$

Equation (12) is the basic recurrence for the Schur-like algorithm; for $-(i-1) \leq j \leq (i-1)$, $s(i, N_1, M_1; j, N_2, M_2) = 0$ by (2).

Setting $j = (i - \frac{1}{2})$ and $-(i - \frac{1}{2})$ in (12) respectively, we can solve for V_i^+ and V_i^- using the following matrix equation

$$\begin{bmatrix} \tilde{S}_i^+ & | & \tilde{S}_i^- \end{bmatrix} = \begin{bmatrix} V_i^+ & | & V_i^- \end{bmatrix} \begin{bmatrix} S_i^{++} & | & S_i^{+-} \\ \text{---} & \text{---} & \text{---} \\ S_i^{-+} & | & S_i^{--} \end{bmatrix} \quad (13)$$

where we have defined the $NM \times NM$ matrices

$$[S_i^{\pm\pm}]_{L_1, L_2} = s(\pm(i - \frac{1}{2}), N_1, M_1; \pm(i - \frac{1}{2}), N_2, M_2) \quad (14)$$

$$[V_i^\pm]_{L_1, L_2} = V_i^\pm(N_1, M_1; N_2, M_2) \quad (15)$$

$$\begin{aligned} [\tilde{S}_i^\pm]_{L_1, L_2} &= s(i - \frac{1}{2}, N_1, M_1; \pm(i - \frac{1}{2}), N_2, M_2) - s(i, N_1, M_1; \pm i, N_2, M_2) \\ &\quad + (\Delta_\theta + \Delta\phi)s(i - \frac{1}{2}, N_1, M_1; \pm(i - \frac{1}{2}), N_2, M_2) \end{aligned} \quad (16)$$

and L_1, L_2 are related to N_1, M_1, N_2, M_2 by

$$L_1 = (N_1 - 1)M + M_1 \quad (17)$$

$$L_2 = (N_2 - 1)M + M_2 \quad (18)$$

for all $1 \leq N_1, N_2 \leq N$, $1 \leq M_1, M_2 \leq M$, and $1 \leq L_1, L_2 \leq NM$

D. Summary of Overall Procedure

The overall procedure can be summarized as follows. Let I_{\max} be the largest radius (maximum radial prediction order). Then for all $1 \leq N_1, N_2 \leq N$ and $1 \leq M_1, M_2 \leq M$:

1. Initialization

Compute $h(\pm\frac{1}{2}, N_1, M_1; 0, N_2, M_2)$, $h(\pm 1, N_1, M_1; 0, N_2, M_2)$ using (2).

Compute $s(\pm\frac{1}{2}, N_1, M_1; j, N_2, M_2)$, $s(\pm 1, N_1, M_1; j, N_2, M_2)$ using (11) for all $j = \pm 1, \dots, \pm 2I_{\max}$.

2. Propagation of Split Schur-Like Algorithm

A Compute the potentials $V_i^+(N_1, M_1; N_2, M_2)$ and $V_i^-(N_1, M_1; N_2, M_2)$ by solving the matrix equation (13);

B Update the Schur variables using (12) for $j = \pm(i + \frac{1}{2}), \dots, \pm 2I_{\max}$.

3. Propagation of Split Levinson-Like Recurrence

A. Propagate the Boundary Points:

$$h(i + \frac{1}{2}, N_1, M_1; i - \frac{1}{2}, N_2, M_2) = h(i, N_1, M_1; i - 1, N_2, M_2) - V_i^+(N_1, M_1; N_2, M_2) \quad (19)$$

$$h(i + \frac{1}{2}, N_1, M_1; -i + \frac{1}{2}, N_2, M_2) = h(i, N_1, M_1; -i + 1, N_2, M_2) - V_i^-(N_1, M_1; N_2, M_2) \quad (20)$$

B. Propagate Non-Boundary Points:

Update $h(i, N_1, M_1; j, N_2, M_2)$ using equation (8) for $j = -(i - \frac{1}{2})$ to $j = (i - \frac{1}{2})$.

4. **Repeat** steps 2 and 3 from $i = 1$ to I_{\max} with increment $\frac{1}{2}$.

The overall procedure is similar to the that for the 2-D algorithm, except that now there are two instead of one angular variables needed to be propagated. The number of MADs required is roughly $O(I_{\max}^2 N^2 M^2)$, which is far less than $O(I_{\max}^2 N^2 M^3)$ MADs required for the generalized Merchant-Parks algorithm and $O(I_{\max}^3 N^3 M^3)$ MADs for Gaussian elimination. Hence, the computational savings in the 3-D case are even more significant than those in the 2-D case. Furthermore, these algorithms are also highly parallelizable, making them even more favorable in the parallel processor environment.

APPENDIX E

W.-H. Fang and A.E. Yagle, "Fast Algorithms for Linear Least-Squares Smoothing Problems in One and Two Dimensions using Generalized Discrete Bellman-Siebert-Krein Resolvent Identities," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992.

Fast Algorithms for Linear Least-Squares Smoothing Problems in One and Two Dimensions Using Generalized Discrete Bellman-Siebert-Krein Resolvent Identities

Wen-Hsien Fang and Andrew E. Yagle
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-2122

Revised October 1990

Abstract

New fast algorithms for linear least-squares smoothing problems in one and two dimensions are derived. These are discrete and multidimensional generalizations of the Bellman-Siebert-Krein resolvent identity, which has been applied to the continuous, one-dimensional stationary smoothing problem by Kailath. The new equations relate the linear least-squares prediction filters associated with discrete random fields to the smoothing filters for those fields. This results in new fast algorithms for deriving the latter from the former. In particular, used in conjunction with recently-developed generalized one (two) dimensional split Levinson and Schur algorithms for covariances with (block) Toeplitz-plus-Hankel structure, these algorithms can be used to compute smoothing filters for random fields defined on a polar raster, using fewer computations than those required by previous algorithms.

I INTRODUCTION

In tomographic imaging problems solved by filtered back-projection [1], and in spotlight synthetic aperture radar [2], data are acquired on a polar raster of points, rather than on a rectangular lattice. Although it is possible to interpolate from the polar raster to a rectangular lattice, it is clearly preferable to deal with the data as it is. This is particularly true if the data are noisy, and smoothing is required.

Regarding the data as a random field with a known covariance function, linear least-squares smoothing may be performed. Computation of the smoothing filter requires solution of two-dimensional discrete normal equations in polar coordinates. Fast algorithms for solving these equations are desirable when the covariance has some structure. However, properties such as stationarity are not manifested as block-Toeplitz structure when the random field is defined on a polar raster. For example, the covariance of an isotropic random field on a rectangular lattice is a Toeplitz function of the abscissae and ordinates, while on a polar raster it is a Toeplitz-plus-Hankel function of the radii.

Kailath [3] has noted the applicability of the Bellman-Siebert-Krein (BSK) resolvent identity to smoothing problems for continuous one-dimensional stationary random processes. First, the *prediction* filter for the process is computed, using the continuous-time Krein-Levinson equations, or by direct solution of the Wiener-Hopf integral equation. Then the BSK identity is used to compute the *smoothing* filter, which is the Fredholm resolvent to the integral operator associated with the covariance function. This approach has been extended to continuous-time close-to-Toeplitz covariances [4] and continuous-parameter isotropic random fields [5], although the latter uses a Fourier expansion into one-dimensional processes.

In this paper we generalize Kailath's approach in three ways: (1) from continuous time to discrete time, resulting in an algorithm directly applicable to real discrete data; (2) from one dimension to two dimensions, without requiring an assumption of isotropy or an initial Fourier expansion; and (3) from stationary to non-stationary random fields.

Although the new algorithms of this paper do NOT require the covariance function to have special structure, they are most useful when used in conjunction with fast algorithms for computing the prediction filters that DO require and exploit special structure in the covariance function. These include the Levinson algorithm [6] for stationary one-dimensional random processes, the algorithm of [7] for non-stationary one-dimensional random processes with Toeplitz-plus-Hankel covariances, and the algorithm of [8] for two-dimensional random fields on a polar raster with Toeplitz-plus-Hankel structure in the radial and angular variables of the covariance.

The paper is organized as follows. Section II derives the algorithm for computing the smoothing filters from the prediction filters for one-dimensional random processes. Section III derives the corresponding algorithm for two-dimensional random fields on a polar raster. Section IV discusses computational complexity, and compares the proposed algorithms to other algorithms for computing the smoothing filters. We also note how the discrete-time equations of this paper reduce to the continuous-time equations of [3] and [9]. Section V concludes with a summary.

II DERIVATION OF THE 1-D SMOOTHING FILTER

A. The Basic Problem

The smoothing problem considered in this section is as follows. Given noisy observations $\{y_k, -M \leq k \leq M\}$ of a zero-mean real-valued random process $\{x_k\}$, compute the linear least-squares estimate of x_k for each k using all of the observations. The observations are related to the process by $y_k = x_k + n_k$, where $\{n_k\}$ is zero-mean discrete white noise with unit power uncorrelated with $\{x_k\}$ (white noise with arbitrary power σ^2 can easily be handled by scaling). The covariance function $k_{i,j} \triangleq E[x_i x_j]$ of $\{x_k\}$ is known, and is assumed to be positive semi-definite.

The linear least-squares estimate \hat{x}_i of x_i based on $\{y_k, -M \leq k \leq M\}$ can be expressed as

$$\hat{x}_i = \sum_{j=-M}^M g_{i,j}^M y_j \quad (1)$$

where the superscript M for $g_{i,j}^M$ denotes that the range of the data is from y_{-M} to y_M . Using the

orthogonality principle of linear least-squares estimation, the *smoothing* filters $g_{i,j}^M$ can be computed by solving the discrete normal equations

$$k_{i,j} = g_{i,j}^M + \sum_{n=-M}^M g_{i,n}^M k_{n,j} \quad \text{for } -M \leq i, j \leq M \quad (2)$$

In the special case when $i = M + 1$, equation (2) becomes the discrete Wiener-Hopf equation

$$k_{M+1,j} = g_{M+1,j}^M + \sum_{n=-M}^M g_{M+1,n}^M k_{n,j} = h_{M+1,j} + \sum_{n=-M}^M h_{M+1,n} k_{n,j} \quad \text{for } -M \leq j \leq M \quad (3)$$

where $h_{M+1,j} \triangleq g_{M+1,j}^M$ is the *prediction* filter. The $h_{i,j}$ are assumed to have been already computed, presumably using some fast algorithm such as those of [6], [7], or [8]. Our objective is to derive a recursive formula for computing the smoothing filters $g_{i,j}^M$ from the previously computed prediction filters $h_{i,j}$.

B. Derivation of the Algorithm

Writing (2) with M replaced by $M + 1$ and subtracting (2) gives

$$0 = (g_{i,j}^{M+1} - g_{i,j}^M) + \sum_{n=-M}^M [(g_{i,n}^{M+1} - g_{i,n}^M) k_{n,j} + g_{i,(M+1)}^{M+1} k_{(M+1),j}] + g_{i,-(M+1)}^{M+1} k_{-(M+1),j} \quad (4)$$

Inserting (3) in (4) results in

$$\begin{aligned} 0 = & (g_{i,j}^{M+1} - g_{i,j}^M) + \sum_{n=-M}^M (g_{i,n}^{M+1} - g_{i,n}^M) k_{n,j} \\ & + g_{i,(M+1)}^{M+1} [h_{(M+1),j} + \sum_{n=-M}^M h_{(M+1),n} k_{n,j}] + g_{i,-(M+1)}^{M+1} [h_{-(M+1),j} + \sum_{n=-M}^M h_{-(M+1),n} k_{n,j}] \end{aligned} \quad (5)$$

and reordering (5) gives

$$\begin{aligned} & (g_{i,j}^{M+1} - g_{i,j}^M) + \sum_{n=-M}^M (g_{i,n}^{M+1} - g_{i,n}^M) k_{n,j} \\ = & -[g_{i,(M+1)}^{M+1} h_{(M+1),j} + g_{i,-(M+1)}^{M+1} h_{-(M+1),j}] - \sum_{n=-M}^M [g_{i,(M+1)}^{M+1} h_{(M+1),n} + g_{i,-(M+1)}^{M+1} h_{-(M+1),n}] k_{n,j} \end{aligned} \quad (6)$$

Since the covariance function $k_{i,j}$ is positive semi-definite by assumption, $\delta_{i,j} + k_{i,j}$ is positive definite, and the solution to any system of equations with system matrix consisting of $\delta_{i,j} + k_{i,j}$ must

be unique. Therefore, we have

$$(g_{i,j}^{M+1} - g_{i,j}^M) = -[g_{i,(M+1)}^{M+1} h_{(M+1),j} + g_{i,-(M+1)}^{M+1} h_{-(M+1),j}] \text{ for all } -M \leq i, j \leq M \quad (7)$$

Equation (7) allows $g_{i,j}^{M+1}$ to be computed recursively from $g_{i,j}^M$ and the prediction filters $h_{\pm(M+1),j}$. Note that $g_{i,j}^M$ and $h_{\pm(M+1),j}$ may be computed in parallel.

C. Computation of Boundary Points

In order to use (7), the boundary points $g_{i,\pm(M+1)}^{M+1}$ must be computed first. This can be done as follows. Setting $j = \pm(M+1)$ in (2), we have

$$\begin{aligned} & [1 + k_{(M+1),(M+1)} - \sum_{n=-M}^M h_{(M+1),n} k_{n,(M+1)}] g_{i,(M+1)}^{M+1} \\ & + [k_{-(M+1),(M+1)} - \sum_{n=-M}^M h_{-(M+1),n} k_{n,(M+1)}] g_{i,-(M+1)}^{M+1} = k_{i,(M+1)} - \sum_{n=-M}^M g_{i,n}^M k_{n,(M+1)} \quad (8) \end{aligned}$$

$$\begin{aligned} & [k_{(M+1),-(M+1)} - \sum_{n=-M}^M h_{(M+1),n} k_{n,-(M+1)}] g_{i,(M+1)}^{M+1} \\ & + [1 + k_{-(M+1),-(M+1)} - \sum_{n=-M}^M h_{-(M+1),n} k_{n,-(M+1)}] g_{i,-(M+1)}^{M+1} = k_{i,-(M+1)} - \sum_{n=-M}^M g_{i,n}^M k_{n,-(M+1)} \quad (9) \end{aligned}$$

These equations can be written as a 2×2 matrix equation for each of the unknown $g_{i,\pm(M+1)}^{M+1}$:

$$\begin{bmatrix} 1 + k_{(M+1),(M+1)} - \sum_{n=-M}^M h_{(M+1),n} k_{n,(M+1)} & k_{-(M+1),(M+1)} - \sum_{n=-M}^M h_{-(M+1),n} k_{n,(M+1)} \\ k_{(M+1),-(M+1)} - \sum_{n=-M}^M h_{(M+1),n} k_{n,-(M+1)} & 1 + k_{-(M+1),-(M+1)} - \sum_{n=-M}^M h_{-(M+1),n} k_{n,-(M+1)} \end{bmatrix} \times \begin{bmatrix} g_{i,(M+1)}^{M+1} \\ g_{i,-(M+1)}^{M+1} \end{bmatrix} = \begin{bmatrix} k_{i,(M+1)} - \sum_{n=-M}^M g_{i,n}^M k_{n,(M+1)} \\ k_{i,-(M+1)} - \sum_{n=-M}^M g_{i,n}^M k_{n,-(M+1)} \end{bmatrix} \quad -(M+1) \leq i \leq (M+1) \quad (10)$$

where we have used the identities $h_{(M+1),n} = g_{(M+1),n}^M$ and $h_{-(M+1),n} = g_{-(M+1),n}^M$. Note that the system matrix in (10) is independent of i .

D. Summary of 1-D Algorithm

Given the data $\{y_k\}$ in the interval $[-L, L]$, the entire algorithm for computing the smoothing filters may be summarized as follows:

1. **Initialize** using $g_{i,j}^{|i|} = h_{i,j}$ for all $-(|i| - 1) \leq j \leq |i| - 1$.
2. Given $g_{i,j}^M$, $-M \leq i, j \leq M$, **update** to $g_{i,j}^{M+1}$ as follows:
 - (a) Compute the boundary points $g_{i,(M+1)}^{M+1}$ and $g_{i,-(M+1)}^{M+1}$ by solving the 2×2 system (10).
 - (b) For each i and j , $-M \leq i, j \leq M$, compute $g_{i,j}^{M+1}$ from $g_{i,j}^M$ using (7). If $k_{i,j}$ has special structure, compute $h_{i,j}$ in parallel using a fast algorithm (e.g., those of [6] or [7]).
 - (c) Continue for $M = |i| - 1$ to L .

III DERIVATION OF THE 2-D SMOOTHING FILTER ON A POLAR RASTER

A. The Basic Problem

Now we consider the smoothing problem for a two-dimensional random field defined on a polar raster, whose points lie along radial lines in $2N$ angular directions (see Fig. 1). The problem considered is as follows. Given noisy observations $\{y_{i,k}, 0 \leq i \leq M, 1 \leq k \leq 2N\}$ of a zero-mean real-valued discrete random field $\{x_{i,k}\}$ at the points (i,k) of a polar raster on a disk, compute the linear least-squares estimate of $x_{i,k}$ for each (i,k) using all of the observations. Here the first subscript denotes radial distance from the origin and the second subscript denotes angular position (k corresponds to the angle $2\pi k/2N$).

The observations $\{y_{i,k}\}$ are related to the random field $\{x_{i,k}\}$ by $y_{i,k} = x_{i,k} + n_{i,k}$, where $\{n_{i,k}\}$ is a zero-mean two-dimensional discrete white noise with unit power uncorrelated with $\{x_{i,k}\}$ (white noise with arbitrary power σ^2 can easily be handled by scaling). The covariance function $k_{i,N_1,j,N_2} \triangleq E[x_{i,N_1} x_{j,N_2}]$ of $\{x_{i,k}\}$ is known, and is assumed to be positive semi-definite.

From Fig. 1, it is clear that the point $(i,k) = (-i, k \pm N)$; in the sequel the point $(i,k), N+1 \leq k \leq 2N$ will be denoted by $(-i, k-N)$. The linear least squares estimate \hat{x}_{i,N_1} of x_{i,N_1} based on $\{y_{j,k}, 0 \leq j \leq M,$

$1 \leq k \leq 2N\} = \{y_{j,k}, -M \leq j \leq M, 1 \leq k \leq N\}$ can be expressed as

$$\hat{x}_{i,N_1} = \sum_{j=-M}^M \sum_{N_2=1}^N g_{i,N_1;j,N_2}^M y_{j,N_2} \quad (11)$$

where the *smoothing* filters $g_{i,N_1;j,N_2}^M$ satisfy the two-dimensional discrete normal equations

$$k_{i,N_1;j,N_2} = g_{i,N_1;j,N_2}^M + \sum_{n=-M}^M \sum_{N_3=1}^N g_{i,N_1;n,N_3}^M k_{n,N_3;j,N_2} \quad -M \leq i,j \leq M, 1 \leq N_1, N_2 \leq N \quad (12)$$

A radial weighting n can be introduced into the double sums in (11) and (12) by replacing $k_{i,N_1;j,N_2}$ and $g_{i,N_1;j,N_2}^M$ with $\sqrt{ij}k_{i,N_1;j,N_2}$ and $\sqrt{ij}g_{i,N_1;j,N_2}^M$; this allows the algorithm to be applied to a discretized two-dimensional integral equation.

B. Derivation of the Algorithm

The derivation is identical to that for the one-dimensional case, since the angular sum is unaffected by the increase of the radial sum from M to $M+1$. The result is (compare to (7)):

$$g_{i,N_1;j,N_2}^{M+1} - g_{i,N_1;j,N_2}^M = -\left[\sum_{N_3=1}^N g_{i,N_1;M+1,N_3}^{M+1} h_{M+1,N_3;j,N_2} + \sum_{N_3=1}^N g_{i,N_1;-(M+1),N_3}^{M+1} h_{-(M+1),N_3;j,N_2} \right] \quad (13)$$

for all $-M \leq i,j \leq M, 1 \leq N_1, N_2 \leq N$

Here the $h_{i,N_1;j,N_2} \triangleq g_{i,N_1;j,N_2}^{|i|-1}$ are the two-dimensional *prediction* filters. The $h_{i,N_1;j,N_2}$ could be computed recursively in parallel with (13), using the fast algorithm of [8].

C. Computation of Boundary Points

As before, we need to compute the boundary points $g_{i,N_1;\pm(M+1),N_2}^{(M+1)}$ prior to using (13). Setting $j = \pm(M+1)$ in (12) results in the equations (compare to (8) and (9))

$$\begin{aligned} & [g_{i,N_1;M+1,N_2}^{M+1} + \sum_{N_3=1}^N g_{i,N_1;M+1,N_3}^{M+1} k_{M+1,N_3;M+1,N_2} - \sum_{N_4=1}^N g_{i,N_1;M+1,N_4}^{M+1} \left(\sum_{n=-M}^M \sum_{N_3=1}^N h_{M+1,N_4;n,N_3} k_{n,N_3;M+1,N_2} \right)] \\ & + \left[\sum_{N_3=1}^N g_{i,N_1;-(M+1),N_3}^{M+1} k_{-(M+1),N_3;M+1,N_2} - \sum_{N_4=1}^N g_{i,N_1;-(M+1),N_4}^{M+1} \left(\sum_{n=-M}^M \sum_{N_3=1}^N h_{-(M+1),N_4;n,N_3} k_{n,N_3;(M+1),N_2} \right) \right] \\ & = k_{i,N_1;(M+1),N_2} - \sum_{n=-M}^M \sum_{N_3=1}^N g_{i,N_1;n,N_3}^M k_{n,N_3;(M+1),N_2} \end{aligned} \quad (14)$$

and

$$\begin{aligned}
& \left[\sum_{N_3=1}^N g_{i,N_1;M+1,N_3}^{M+1} k_{M+1,N_3;-(M+1),N_2} - \sum_{N_4=1}^N g_{i,N_1;-(M+1),N_4}^{M+1} \left(\sum_{n=-M}^M \sum_{N_3=1}^N h_{M+1,N_4;n,N_3} k_{n,N_3;-(M+1),N_2} \right) \right] \\
& + [g_{i,N_1;-(M+1),N_2}^{(M+1)} + \sum_{N_3=1}^N g_{i,N_1;-(M+1),N_3}^{(M+1)} k_{-(M+1),N_3;-(M+1),N_2} \\
& - \sum_{N_4=1}^N g_{i,N_1;-(M+1),N_4}^{(M+1)} \left(\sum_{n=-M}^M \sum_{N_3=1}^N h_{-(M+1),N_4;n,N_3} k_{n,N_3;-(M+1),N_2} \right)] \\
& = k_{i,N_1;-(M+1),N_2} - \sum_{n=-M}^M \sum_{N_3=1}^N g_{i,N_1;n,N_3}^M k_{n,N_3;-(M+1),N_2} \quad (15)
\end{aligned}$$

If we define the following $N \times N$ matrices ($1 \leq N_1, N_2 \leq N$)

$$[G^\pm]_{N_1, N_2} \triangleq g_{i, N_1; \pm(M+1), N_2}^{M+1} \quad (16)$$

$$[K^{\pm, \pm}]_{N_1, N_2} \triangleq k_{\pm(M+1), N_1; \pm(M+1), N_2} \quad (17)$$

$$[H^\pm K^\pm]_{N_1, N_2} \triangleq \sum_{n=-M}^M \sum_{N_3=1}^N h_{\pm(M+1), N_1; n, N_3} k_{n, N_3; \pm(M+1), N_2} \quad (18)$$

and then define from equations (16)-(18) the additional $N \times N$ matrices

$$A \triangleq I + K^{++} - H^+ K^+, \quad B \triangleq K^{-+} - H^- K^+ \quad (19)$$

$$C \triangleq K^{+-} - H^+ K^-, \quad D \triangleq I + K^{--} - H^- K^- \quad (20)$$

$$[R]_{N_1, N_2} \triangleq k_{i, N_1; (M+1), N_2} - \sum_{n=-M}^M \sum_{N_3=1}^N g_{i, N_1; n, N_3}^M k_{n, N_3; (M+1), N_2} \quad (21)$$

$$[S]_{N_1, N_2} \triangleq k_{i, N_1; -(M+1), N_2} - \sum_{n=-M}^M \sum_{N_3=1}^N g_{i, N_1; n, N_3}^M k_{n, N_3; -(M+1), N_2} \quad (22)$$

then equations (14) and (15) can be written in matrix form as

$$G^+ A + G^- B = R \quad (23)$$

$$G^+ C + G^- D = S \quad (24)$$

Equations (23) and (24) are a $2N \times 2N$ system of equations for G^+ and G^- ; compare them with (10) (for which $N = 1$). However, a further simplification is possible. Since the system matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is the same for each i , equations (23) and (24) can be solved in closed form to give

$$G^+ = (R - SD^{-1}B)(A - CD^{-1}B)^{-1} \quad (25)$$

$$G^- = (S - RA^{-1}C)(D - BA^{-1}C)^{-1} \quad (26)$$

independent of i .

Hence computation of the boundary points $g_{i,N_1;\pm(M+1),N_2}^{M+1}$ for all i requires only the inversion of four $N \times N$ matrices in (25) and (26). This is significant, since the smoothing filters $g_{i,N_1;j,N_2}^M$ will generally be computed for all i and N_1, N_2 (we generally wish to smooth all or most of an image, not just one pixel). This is where our algorithm saves a significant amount of computation, as compared with other algorithms (see below).

D. Summary of 2-D Algorithm

Given the data $\{y_{i,k}, -L \leq i \leq L, 0 \leq k \leq N\}$, the entire algorithm for computing the two-dimensional smoothing filters may be summarized as follows:

1. **Initialize** using $g_{i,N_1;j,N_2}^{|i|-1} = h_{i,N_1;j,N_2}$ for all $-(|i| - 1) \leq j \leq |i| - 1$ and $1 \leq N_1, N_2 \leq N$.
2. Given $g_{i,N_1;j,N_2}^M$, $-M \leq i, j \leq M, 1 \leq N_1, N_2 \leq N$, **update** to $g_{i,N_1;j,N_2}^{M+1}$ as follows:
 - (a) Compute the boundary points $g_{i,N_1;(M+1),N_2}^{M+1}$ and $g_{i,N_1;-(M+1),N_2}^{M+1}$ by solving in parallel the $2M + 1$ $2N \times 2N$ systems (25) and (26).
 - (b) For each i and j , $-M \leq i, j \leq M$, and each N_1 and $N_2, 1 \leq N_1, N_2 \leq N$, compute $g_{i,N_1;j,N_2}^{M+1}$ from $g_{i,N_1;j,N_2}^M$ using (13). If $h_{i,N_1;j,N_2}$ has special structure, compute $h_{i,N_1;j,N_2}$ in parallel using a fast algorithm (e.g. the algorithm of [8]).
 - (c) Continue for $M = |i| - 1$ to L .

IV COMPUTATIONAL COMPLEXITY

We determine the number of Multiplications-And-Divisions (MADs) needed to compute the smoothing filters from the prediction filters. We also determine the total number of MADs needed to compute the smoothing filters from the covariance function, assuming that the latter has special structure and a fast algorithm has been used to compute the prediction filters. Although some current DSP chips can perform multiplications as quickly as additions, the fact remains that multiplication is a more complex operation than addition. MADs can still be used as a rough guide to the computational complexity of an algorithm.

A. Computational Complexity of the One-Dimensional Algorithm

The number of MADs needed to compute the smoothing filters from the prediction filters, given data $\{y_j, -L \leq j \leq L\}$, can be determined as follows. For each i , updating the smoothing filters from $g_{i,j}^M$ to $g_{i,j}^{M+1}$ (this corresponds to adding two data points at $j = M + 1$ and $j = -(M + 1)$) requires $6(2M + 1) + 8$ MADs to compute the boundary points $g_{i,\pm(M+1)}^{M+1}$ (the six sum-of-products computations in (10)), and $2(2M + 1)$ MADs to update the other $g_{i,j}^M$ to $g_{i,j}^{M+1}$ in (7). The total number of MADs to compute $g_{i,j}^L$ for *one* i and all j is thus $\sum_{M=|i|-1}^L [8(2M + 1) + 8] = 8(L^2 - i^2) + 24L + 8|i| + 16$. However, the total number of MADs needed to compute $g_{i,j}^L$ for *all* i and j is only $\sum_{M=1}^L [4(2M + 1) + 2] + \sum_{i=-L}^L \sum_{M=|i|-1}^L [4(2M + 1) + 6] = 5\frac{1}{3}L^3 + 34L^2 + 48\frac{2}{3}L + 12$, since the system matrix in (10) is independent of i , and thus need not be re-computed and re-inverted for each i .

In the sequel, we assume (for purposes of comparison) that $L \gg 1$ and $i \gg 1$. Then the dominant terms in the number of MADs are the terms of highest order in L and i . To facilitate comparisons, only these dominant terms will be given.

If the covariance $k_{i,j}$ is Toeplitz, i.e. $\{x_k\}$ is a stationary process, then we have $k_{i,j} = k_{|i-j|} = k_{-i,-j}$ and $g_{i,j}^M = g_{-i,-j}^M$ from (2). Then two of the four sum-of-product computations in the system matrix of (10) are redundant, so that computation of $g_{i,j}^L$ for one i and all j requires only $6(L^2 - i^2)$ MADs. Also,

Covariance	Filter for	LTZ or MP	L+BSK or [7]+BSK
Symmetric	single point i	$8L^2$	$10L^2 - 6i^2$
Toeplitz	$all -L \leq i \leq L$	$4L^3$	$2\frac{2}{3}L^3$
Toeplitz-plus-	single point i	$64L^2$	$32L^2 - 8i^2$
Hankel	$all -L \leq i \leq L$	$64L^3$	$5\frac{1}{3}L^3$

Table 1: Numbers of MADs required for some specific covariance functions to solve (2)

Covariance	Filter for	LWR or MP	LWR+BSK or [8]+BSK
Block	single point i	$10L^2N^3$	$(14L^2 - 8i^2)N^3$
Toeplitz	$all -L \leq i \leq L$	$8L^3N^3$	$5\frac{1}{3}L^3N^3$
Block-Toeplitz	single point i	$64L^2N^3$	$8(L^2 - i^2)N^3$
-plus-Hankel	$all -L \leq i \leq L$	$64L^3N^3$	$5\frac{1}{3}L^3N^3$

Table 2: Numbers of MADs required for some specific covariance functions to solve (12)

since $g_{i,j}^L$ need only be computed for $i \geq 0$, computation of $g_{i,j}^L$ for all i and j requires only half as many MADs as before, viz. $2\frac{2}{3}L^3$. Furthermore, the Levinson algorithm (L) [6] may be used to compute the prediction filters $\{h_{i,j}, -L \leq i, j \leq L\}$ from $k_{i,j}$, at a cost of $4L^2$ MADs. The Levinson algorithm can be propagated in parallel with our algorithm, resulting in an overall fast algorithm for computing the smoothing filters $g_{i,j}^L$ from $k_{i,j}$. If the covariance is Toeplitz-plus-Hankel, the fast algorithm of [7] may be used to compute the $\{h_{i,j}, -L \leq i, j \leq L\}$ from the $k_{i,j}$, at a cost of $24L^2$ MADs, again in parallel with our algorithm. However, we no longer have $g_{i,j}^M = g_{-i,-j}^M$, so the reductions in computation for purely Toeplitz covariances no longer apply.

The major alternatives to these procedures are the Levinson-Trench-Zohar (LTZ) [10] algorithm for Toeplitz systems, and the algorithm of Merchant and Parks (MP) [11] for Toeplitz-plus-Hankel systems. We compare the numbers of MADs required by all of these algorithms in Table 1.

For a Toeplitz covariance, it can be seen from Table 1 that if $g_{i,j}^L$ for a *single* point i is desired, i.e., we wish to compute a smoothed estimate at only *one* point, then the LTZ algorithm is superior to ours for small values of i , while ours is superior for large values of i . However, if $g_{i,j}^L$ for *all* points $-L \leq i \leq L$ is desired, i.e., we wish to compute smoothed estimates at *all* points (as would generally be the case), then our algorithm in conjunction with the Levinson algorithm requires only $\frac{2}{3}$ as many MADs for large L . Furthermore, for Toeplitz-plus-Hankel covariances, our algorithm in conjunction with that of [7] requires less than half as many MADs to compute $g_{i,j}^L$ for a *single* point i , and $\frac{1}{12}$ as many MADs to compute $g_{i,j}^L$ for *all* i when L is large. Further savings are possible since many computations (e.g., the updates and the sum in (10)) can be done in parallel.

Other approaches may require still more computation. $g_{i,j}^M$ may be updated to $g_{i,j}^{M+1}$ using the well-known formula for updating the inverse of a partitioned matrix. However, this requires $3M^2$ MADs per update, as opposed to the $8(2M+1) + 8$ MADs required by the BSK identity. Direct solution of (2) using Gaussian elimination would require $\frac{1}{3}(2L+1)^3 + \frac{1}{2}(2L+1)^2$ MADs for each i .

B. Computational Complexity of the Two-Dimensional Algorithm

We now assume that the observations are $\{y_{j,k}, -L \leq j \leq L, 1 \leq k \leq N\}$, so that updating the smoothing filters from $g_{i,N_1;j,N_2}^M$ to $g_{i,N_1;j,N_2}^{M+1}$ corresponds to adding a "shell" of $2N$ data points at radius $M+1$. For each i , computation of the boundary points $g_{i,N_1;\pm(M+1),N_2}^{M+1}$ requires $6(2M+1)N^3$ MADs for (16)-(22), and $4N \times N$ matrix multiplications and inversions for (25) and (26). Updating the other smoothing filters from $g_{i,N_1;j,N_2}^M$ to $g_{i,N_1;j,N_2}^{M+1}$ requires $2(2M+1)N^3$ MADs for (13). Hence the number of MADs needed to compute $g_{i,N_1;j,N_2}^L$ from the prediction filters for *one* i and all j is $8(L^2 - i^2)N^3$, while the number of MADs needed for *all* i and j is $5\frac{1}{3}L^3N^3$. Note that these are the numbers for the one-dimensional algorithm multiplied by N^3 , since all operations now involve matrices.

In the sequel, we assume (for purpose of comparison) that $L \gg N \gg 1$. If the covariance $k_{i,N_1;j,N_2}$ is block-Toeplitz, i.e. Toeplitz in i and j , then the Levinson-Wiggins-Robinson (LWR) [12]

algorithm may be used to compute the prediction filters h_{i,N_1,j,N_2} from k_{i,N_1,j,N_2} , at a cost of $6L^2N^3$ MADs (recall that the backward predictors are no longer the time-reversed forward predictors, in the multichannel case). If the covariance is Toeplitz-plus-Hankel in both i and j and N_1 and N_2 , as it is for an isotropic random field on a polar raster, the fast algorithm of [8] may be used to compute the prediction filters, at a cost of $24L^2N^2$ MADs.

The major alternatives to these procedures are the LWR algorithm adapted to an arbitrary block-Toeplitz system, and a matrix generalization of the Merchants-Parks procedure for block Toeplitz-plus-Hankel systems. Results are summarized in Table 2. The savings are similar to those for the one-dimensional algorithms, except for the even greater savings for block Toeplitz-plus-Hankel covariances. The reason for the great savings here is the efficiency of the algorithm of [8], which requires only $24L^2N^2$ MADs to determine the prediction filters from the covariance function; that is *negligible* compared to $8(L^2 - i^2)N^3$ and $5\frac{1}{3}L^3N^3$ if $L \gg N \gg 1$.

C. Relation to Continuous-Parameter BSK Identities

It is instructive to examine the continuous-parameter limits of the various equations of this paper. Let the intervals between points be δ_r in the radial direction and $\delta_\theta = \frac{2\pi}{N}$ radians in the angular direction. Introducing a radial weighting factor, as discussed below (12), and taking limits as δ_r and δ_θ go to zero results in the following transformations:

1. The discrete normal equations (2) and (12) become Fredholm integral equations. Similarly, the discrete Wiener-Hopf equation (3) and its two-dimensional counterpart become Wiener-Hopf integral equations;
2. The smoothing filters become the Fredholm resolvents to the integral operators associated with the covariance functions;

3. Using $g_{i,M+1}^M = g_{M+1,i}^M = h_{M+1,i}$, equation (7) becomes

$$\begin{aligned}\frac{\partial g}{\partial T}(x, y; T) &= -(g(x, T; T)h(T, y) + g(x, -T; T)h(-T, y)) \\ &= -(h(T, x)h(T, y) + h(-T, x)h(-T, y))\end{aligned}\quad (27)$$

where $g(x, y; T)$ is the smoothing function by which an observation at y in the interval $[-T, T]$ is multiplied and integrated to compute an estimate at x . Equation (27) is the *BSK resolvent identity* (modified from $[0, T]$ to $[-T, T]$), which was applied to continuous-time smoothing problems in [3];

4. Similarly, the recursion (13) becomes

$$\frac{\partial g}{\partial T}(|x|e_x, |y|e_y; T) = - \int_S g(|x|e_x, Te'; T)h(Te', |y|e_y)T^2 de' \quad (28)$$

where e_x, e_y and e' are unit vectors, $x = |x|e_x, y = |y|e_y$ and S is the unit circle. Equation (28) is identical to the generalized BSK identity applied to a multi-dimensional continuous-parameter smoothing problem in [9];

5. Since $\delta_{i,j}$ becomes a continuous-time impulse, the units in (10) and (19)-(20) dominate the other terms. Hence the computations of the boundary points (10) and (25)-(26) become, respectively,

$$g(x, y; T) = k(x, y) - \int_0^T g(x, z; T)k(z, y)dz \quad (29)$$

$$g(|x|e_x, |y|e_y; T) = k(|x|e_x, |y|e_y) - \int_0^T \int_S g(|x|e_x, |z|e_z; T)k(|z|e_z, |y|e_y)|z|de_z d|z| \quad (30)$$

which agree with equations for computing boundary values that appear in [3]-[5] and [9].

Note that although the discrete equations transform into the expected continuous equations, the forms of the discrete equations are not obvious from the continuous equations.

V CONCLUSION

New fast algorithms for computing the linear least-squares smoothing filters for random processes and fields have been derived. These algorithms relate the smoothing filters to the prediction filters

associated with the same covariance. If the covariance has special structure, fast algorithms such as those of [6], [7], and [8] may be used to compute the prediction filters; such algorithms may be propagated in parallel with those of this paper. This can result in significant computational savings. However, it is important to emphasize that the results of this paper hold for *arbitrary* covariances, and do not rely on the existence of such fast algorithms.

In the limit of continuous time, the one-dimensional algorithm reduces to the BSK identity, which was applied previously to smoothing problems for continuous-time stationary random processes by Kailath. However, the algorithms are non-trivial discrete and two-dimensional generalizations of the BSK identity. Since both data and numerical computation are inherently discrete in nature, these algorithms constitute a significant step in the practical application of these smoothing ideas.

ACKNOWLEDGMENT

The work of both authors was supported by the Air Force Office of Scientific Research under grant # AFOSR-89-0017.

References

- [1] G.T. Herman ed., *Image Reconstruction from Projections: Implementation and Applications*, Springer-Verlag, NY, 1979.
- [2] J.L. Walker, *Range-Doppler imaging of rotating objects*, IEEE Trans. Aerosp. Electron. Syst. vol. AES-16, pp. 23-52, Jan. 1980.
- [3] T. Kailath, *Application of a resolvent identity to a linear smoothing problem*, SIAM J. Cont., vol. 7, no. 1, pp. 68-74, Feb. 1969.
- [4] T. Kailath, B. Levy, L. Ljung, and M. Morf, *The factorization and representation of operators in the algebra generated by Toeplitz operators*, SIAM J. Appl. Math., vol. 37, no. 3, pp. 467-484, Dec. 1979.

- [5] B.C. Levy, J.N. Tsitsiklis, *A fast algorithm for linear estimation of two-dimensional isotropic random fields*, IEEE Trans. on Info. Th., vol. IT-31, No. 5, pp.635-644, Sept. 1985.
- [6] N. Levinson, *The Wiener RMS error criterion in filter design and prediction*, J. Math. Phys 25, pp.261-278, June 1947.
- [7] A.E. Yagle, *New analogues of split algorithms for arbitrary Toeplitz-plus-Hankel matrices*, to appear in IEEE Trans. ASSP.
- [8] W.-H. Fang, A.E. Yagle, *Discrete fast algorithms for two-dimensional linear prediction on a polar raster*, submitted to IEEE Trans. ASSP, Oct. 1989.
- [9] A.E. Yagle, *Analogues of split Levinson, Schur, and lattice algorithms for three-dimensional random field estimation problems*, to appear in SIAM J. Appl. Math, Dec. 1990.
- [10] S. Zohar, *The solution of a Toeplitz set of linear equations*, J. Assoc. Comput. Math., vol. 21, pp. 272-276, 1974.
- [11] G.A. Merchant and T.W. Parks, *Efficient solution of a Toeplitz-plus-Hankel coefficient matrix system of equations*, IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-30, no.1, pp. 40-44, Feb. 1982.
- [12] R.A. Wiggins and E.A. Robinson, *Recursive solution to the multichannel filtering problem*, J. Geophys. Res., vol. 70, pp. 1885-1891, Apr. 1965.

FIGURE HEADING

1. The polar raster on which the two-dimensional random field is defined with $2N = 8$.

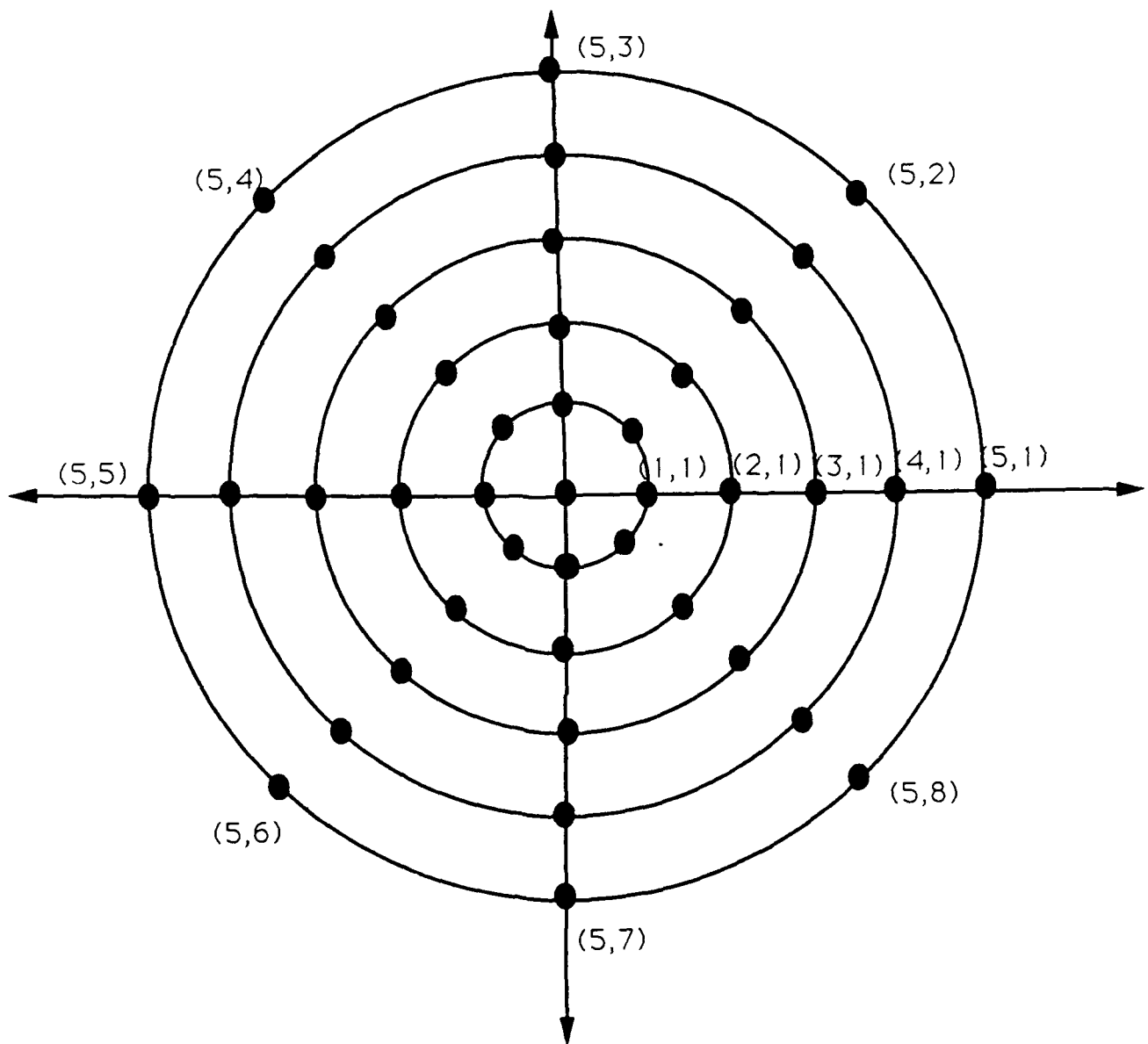


Figure 1: 2-D polar raster with
 $0 \leq \text{radius} \leq 5$; $1 \leq \text{angular part} \leq 8$

APPENDIX F

W.-H. Fang and A.E. Yagle, "Two Methods for Toeplitz-plus-Hankel Approximation to a Data Covariance Matrix," to appear in *IEEE Trans. Signal Processing*, vol. ASSP-40, no. 6, June 1992.

Two Methods for Toeplitz-plus-Hankel Approximation to a Data Covariance Matrix

Wen-Hsien Fang and Andrew E. Yagle
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-2122

Revised March 1991

Abstract

Recently, fast algorithms have been developed for computing the optimal linear least-squares prediction filters for non-stationary random processes (fields) whose covariances have (block) Toeplitz-plus-Hankel form. If the covariance of the random process (field) must be estimated from the data itself, we have the following problem: Given a data covariance matrix, computed from the available data, find the Toeplitz-plus-Hankel matrix closest to this matrix in some sense. This paper gives two procedures for computing the Toeplitz-plus-Hankel matrix that minimizes the Hilbert-Schmidt norm of the difference between the two matrices. The first approach projects the data covariance matrix onto the subspace of Toeplitz-plus-Hankel matrices, for which basis functions can be computed using a Gram-Schmidt orthonormalization. The second approach projects onto the subspace of symmetric Toeplitz plus skew-persymmetric Hankel matrices, resulting in a much simpler algorithm. The extension to block Toeplitz-plus-Hankel data covariance matrix approximation is also addressed.

I INTRODUCTION

Some fast algorithms have recently been developed for computing the optimal linear least-squares prediction filters for non-stationary random processes (fields) whose covariances have (block) Toeplitz-plus-Hankel form [1, 2, 3]. Often the covariance function is not given explicitly, but must be estimated from the data itself. To utilize these fast algorithms, the estimated covariance function must have Toeplitz-plus-Hankel structure. The problem can be posed: Given a data covariance matrix, computed from a data sequence, find a Toeplitz-plus-Hankel matrix that is closest to the data matrix in some sense.

Several common random processes (fields) have (block) Toeplitz-plus-Hankel covariance functions. For example, the first-order Gauss-Markov process

$$x_n = ax_{n-1} + w_n, \quad n \geq 1, \quad x_0 = 0, \quad |a| < 1, \quad (1)$$

where w_n is discrete white noise with variance σ^2 , has the Toeplitz-plus-Hankel covariance function

$$K(i, j) \triangleq E[x_i x_j] = \frac{\sigma^2}{1 - a^2} (a^{|i-j|} - a^{|i+j|}), \quad i, j \geq 0. \quad (2)$$

The two-dimensional circularly symmetric Markovian random field on a polar raster

$$x_{i,N} = a \sum_{n=1}^M x_{i-1,n} + w_{i,N}, \quad i \geq 1, \quad 1 \leq N \leq M, \quad x_{0,N} = 0, \quad |a| < 1, \quad (3)$$

where $(i, 2\pi \frac{N}{M})$, $1 \leq N \leq M$ are polar coordinates on the polar raster and $w_{i,N}$ is two-dimensional white noise with variance $\frac{\sigma^2}{M}$, also has the Toeplitz-plus-Hankel covariance (2). Also, in image processing a two-dimensional isotropic random field is often modelled [4] as having a covariance function

$$\begin{aligned} E[x_{(i, 2\pi \frac{N_1}{M})} x_{(j, 2\pi \frac{N_2}{M})}] &\triangleq K(i, 2\pi \frac{N_1}{M}; j, 2\pi \frac{N_2}{M}) = \rho^{i^2+j^2-2ij\cos(2\pi(N_1-N_2)/M)} \\ &= \rho^{\frac{1}{2}([(i+j)^2+(i-j)^2]-[(i+j)^2-(i-j)^2]\cos(2\pi(N_1-N_2)/M))} \\ &\approx 1 + \frac{1}{2}([(i+j)^2+(i-j)^2]-[(i+j)^2-(i-j)^2]\cos(2\pi(N_1-N_2)/N)) \ln \rho \end{aligned} \quad (4)$$

if $\rho \approx 1$, which has a block Toeplitz-plus-Hankel structure. Clearly, for these and similar random processes (fields), a Toeplitz-plus-Hankel structured covariance estimate will be much more accurate than a Toeplitz estimate.

For the special case of a wide-sense stationary random process, the estimated covariance matrix is symmetric Toeplitz. The matrix minimizing the Hilbert-Schmidt norm of the difference between this matrix and the data covariance matrix is found by averaging the diagonals of the data covariance matrix, replacing each element being averaged by the average [5]. This is the result of projecting the data covariance matrix on the vector space of all symmetric Toeplitz matrices, where the projection is defined using the Hilbert-Schmidt inner product.

In this paper we extend this approach to the more general case of Toeplitz-plus-Hankel matrices, following which the algorithms of [1, 2, 3] may be applied. Since the subspace of symmetric Toeplitz matrices is a subset of the subspace of symmetric Toeplitz-plus-Hankel matrices, the errors (in the Hilbert-Schmidt norm sense) will always be smaller than the error using only the Toeplitz approximation. Unfortunately, the method is more complicated than simply averaging along diagonals as in Toeplitz approximation. The basis elements of the subspace need to be computed using a Gram-Schmidt orthogonalization, and there seems to be no simple closed-form expression for an arbitrary element. However, if we restrict ourselves to the subspace of symmetric Toeplitz plus skew-persymmetric Hankel matrices, the optimal approximation can be easily derived by simply averaging along diagonals and antidiagonals. Both methods are developed in this paper. The extension to approximation for block data covariance matrices is also included. We do not specifically address other constraints such as positive definiteness, although such constraints can be incorporated into one of the methods of Section IV, if needed.

This paper is organized as follows. In Section II, we specify the problem, the criterion used, and the approach employed. In Section III, the optimal Toeplitz-plus-Hankel approximation using basis elements derived from a Gram-Schmidt orthogonalization is derived. In Section IV, the

optimal symmetric Toeplitz plus skew-persymmetric Hankel approximation using averaging along the diagonals and antidiagonals is derived. Some examples are also given to demonstrate the procedures. In Section V, the results are extended to block data covariance matrix approximation. Section VI concludes with a summary.

II PROBLEM FORMULATION

A. Hilbert-Schmidt Norm

For any two square real $n \times n$ matrices A and B , the Hilbert-Schmidt inner product and norm are defined as

$$\langle A, B \rangle \triangleq \text{Trace}[AB^T] ; \quad \|A\|^2 \triangleq \langle A, A \rangle = \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 \quad (5)$$

The problem we will deal with can be posed as follows: Given a data covariance matrix R , find the Toeplitz-plus-Hankel matrix \tilde{R} such that $\|R - \tilde{R}\|$ is minimized.

The solution to this problem can be easily derived by projecting R onto the subspace of Toeplitz-plus-Hankel matrices. A set of matrices spanning this subspace is

$$\begin{bmatrix} 1 & \dots & \dots & 0 \\ \vdots & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \vdots \\ 0 & \dots & \dots & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & \dots & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & \vdots \\ 1 & \dots & \dots & 0 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} 0 & \dots & \dots & 1 \\ \vdots & \vdots & 1 & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \dots & \dots & 0 \end{bmatrix}, \begin{bmatrix} 0 & \dots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix}, \begin{bmatrix} 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \dots & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \quad (7)$$

where the $2n - 1$ basis function in (6) span the Toeplitz matrices, and the $2n - 1$ matrices in (7) span the Hankel matrices.

B. Projection Approach

If we are given a set of orthogonal matrices $\{Q_i\}_{i=1}^k$, then the minimum distance (norm) between a matrix R and the matrix \tilde{R} in the subspace spanned by $\{Q_i\}_{i=1}^k$ is equal to the distance between the matrix R and its projection on this subspace, i.e., if $\|R - \tilde{R}\|$ is minimum, then

$$\tilde{R} = \sum_{i=1}^k \frac{\langle R, Q_i \rangle}{\langle Q_i, Q_i \rangle} Q_i \quad (8)$$

Consider the special case where the $\{Q_i\}_{i=1}^{2n-1}$ are the matrices in (6). Then, since the $\{Q_i\}_{i=1}^{2n-1}$ span the subspace of Toeplitz matrices and are orthogonal in the inner product (5), the optimal Toeplitz approximation for any matrix is to project the matrix on this subspace, and this leads to averaging along diagonals [5]. If we extend the basis $\{Q_i\}$ to include basis elements for Hankel matrices as well, the error metric $\|R - \tilde{R}\|$ will clearly be less than the error for Toeplitz-only approximation. Let \tilde{R}_T be the optimal Toeplitz matrix approximation to R , and let \tilde{R}_{TH} be the optimal Toeplitz-plus-Hankel approximation to R . Then the improvement in the error metric is

$$\|R - \tilde{R}_{TH}\|^2 = \|R - \tilde{R}_T\|^2 - \|\tilde{R}_H\|^2 \quad (9)$$

where \tilde{R}_H is the projection of R on the extension of the basis $\{Q_i\}_{i=1}^{2n-1}$ to include Hankel matrices. We now discuss this basis extension.

III OPTIMAL TOEPLITZ-PLUS-HANKEL APPROXIMATION

A. Gram-Schmidt Orthogonalization

Unfortunately, while the matrices in (6) are orthogonal, and those in (7) are orthogonal, the union of (6) and (7) are not orthogonal in the sense of Hilbert-Schmidt norm defined in (5). So while (6) and (7) span the subspace of Toeplitz-plus-Hankel matrices, they are not an orthogonal basis. Hence the projection of R can not be computed by averaging along the diagonals and antidiagonals.

To use the projection method, the matrices in (7) must be Gram-Schmidt orthogonalized, extending the orthogonal basis in (6). If we represent the matrices in (6) and (7) as $\{Q_i\}_{i=1}^{2n-1}$, and $\{Q_i\}_{i=2n}^{4n-2}$ respectively, then the new orthogonal basis functions Q'_i can be recursively computed by

$$Q'_i = Q_i \quad \text{for } i = 1, \dots, 2n-1 \quad (10)$$

$$Q'_i = Q_i - \sum_{k=1}^{2n-1} \frac{\langle Q_i, Q_k \rangle}{\langle Q_k, Q_k \rangle} Q_k - \sum_{j=2n}^{i-1} \frac{\langle Q_i, Q'_j \rangle}{\langle Q'_j, Q'_j \rangle} Q'_j \quad \text{for } i = 2n, \dots, 4n-2 \quad (11)$$

B. Solution Procedures

Given a data covariance matrix R , the desired Toeplitz-plus-Hankel approximation \tilde{R} can be computed as follows:

1. Adjoin the set of $2n - 1$ Toeplitz orthogonal basis elements in (6) to some additional Hankel orthogonal basis elements computed using the Gram-Schmidt procedure of (10)-(11). This yields a complete orthogonal basis, say $\{Q_1, Q_2, \dots, Q_{4n-4}\}$. (It is shown in Appendix A that there are $4n - 4$ orthogonal matrices in this subspace.)
2. Compute \tilde{R} using (8), with $k = 4n - 4$. The projections on the Toeplitz matrices are found by averaging along diagonals. The projections on the Hankel basis elements are found by taking linear combinations of the element of R as follows:
3. To compute $\langle R, Q_i \rangle$ for the Hankel basis elements $2n \leq i \leq 4n - 4$, regard Q_i as a stencil. Overlay R with Q_i and multiply each element of R by the element of Q_i directly over it. Note that for each Q_i at least half of the elements are zero.

C. Example

$$\text{Let } R = \begin{bmatrix} 6 & 3 & 8 \\ 2 & 1 & 7 \\ 6 & 4 & 11 \end{bmatrix}$$

The optimal Toeplitz-plus-Hankel approximation \tilde{R} can be computed as

$$\begin{aligned} \tilde{R} &= \frac{6+1+11}{3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \frac{3+7}{2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \frac{2+4}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ &+ 8 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 6 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} + \frac{6(-0.33) + 1(0.67) + 11(-0.33)}{0.67} \begin{bmatrix} -0.33 & 0 & 0 \\ 0 & 0.67 & 0 \\ 0 & 0 & -0.33 \end{bmatrix} \\ &+ \frac{3(0.5) - 7(0.5) + 2(0.5) - 4(0.5)}{1} \begin{bmatrix} 0 & 0.5 & 0 \\ 0.5 & 0 & -0.5 \\ 0 & -0.5 & 0 \end{bmatrix} + \frac{6(0.5) - 11(0.5)}{0.5} \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -0.5 \end{bmatrix} \\ &= \begin{bmatrix} 6 & 3.5 & 8 \\ 1.5 & 1 & 6.5 \\ 6 & 4.5 & 11 \end{bmatrix} \end{aligned}$$

The Hilbert-Schmidt norm of the error for this Toeplitz-plus-Hankel approximation is equal to 1. For Toeplitz-only approximation, the error norm is 7.75. The reduction from $\|R - \tilde{R}_T\| = 7.75$ to $\|R - \tilde{R}_{TH}\| = 1$ is due to $\|\tilde{R}_H\| = 7.68$ in (9). Note that the elements of \tilde{R} are very close to those of R . This is not surprising, since for this example $n = 3$, and there are $4n - 4 = 8$ basis functions, only one shy of number of the degrees of freedom required to completely specify an arbitrary 3×3 matrix.

IV OPTIMAL SYMMETRIC TOEPLITZ PLUS SKEW-PERSYMMETRIC HANKEL APPROXIMATION

The major computational complexity of the above method lies in the Gram-Schmidt orthogonalization procedure. We now show that if we restrict ourselves to a specific class of Toeplitz-plus-Hankel matrices, we obtain a much simpler algorithm which involves simply averaging along diagonals and antidiagonals of the data covariance matrix. This is done in two parts: First, we use a matrix identity to transform this special case of the Toeplitz-plus-Hankel approximation into the more familiar Hermitian Toeplitz approximation problem. Second, we show that this problem is equivalent to averaging along diagonals and antidiagonals.

A. Transformation to Hermitian Toeplitz Approximation

For simplicity, we only consider the case where n is even. Define I_n as the $n \times n$ identity matrix, and J_n as the $n \times n$ exchange matrix with ones on the main antidiagonal. It has been shown in [6] that for any $n \times n$ Hermitian Toeplitz matrix HT ,

$$U = \frac{1}{2} \begin{bmatrix} (1-j)I_{\frac{n}{2}} & (1+j)J_{\frac{n}{2}} \\ (1+j)J_{\frac{n}{2}} & (1-j)I_{\frac{n}{2}} \end{bmatrix} ; \quad U^{-1} = U^H = \frac{1}{2} \begin{bmatrix} (1+j)I_{\frac{n}{2}} & (1-j)J_{\frac{n}{2}} \\ (1-j)J_{\frac{n}{2}} & (1+j)I_{\frac{n}{2}} \end{bmatrix} \quad (12)$$

will transform HT into a sum of real Toeplitz and Hankel matrices :

$$\widehat{HT} = U(HT)U^H = T \text{ (Toeplitz matrix)} + H \text{ (Hankel matrix)} \quad (13)$$

where $T = \text{Re}[HT]$ and $H = \text{Im}[HT] \cdot J_n$. Since HT is a Hermitian Toeplitz matrix, T is a

symmetric Toeplitz matrix and H is a skew-persymmetric Hankel matrix with all zero elements on the main antidiagonal.

Since our concern is to obtain the optimal Toeplitz-plus-Hankel approximation, we will reverse the above procedure. More specifically, given any data covariance matrix R , we want to find the optimal Toeplitz-plus-Hankel approximation \tilde{R} , where the Toeplitz and Hankel matrices have the same structure as those of (13). Then

$$\min_{\tilde{R}} \|R - \tilde{R}\| = \min_{\tilde{R}} \|U^H(R - \tilde{R})U\| = \min_{HT} \|U^H RU - HT\| \quad (14)$$

where $HT \triangleq U^H \tilde{R} U$ is a Hermitian Toeplitz matrix, and we have used the fact that unitary transformation is a one-to-one mapping that does not change the Hilbert-Schmidt norm.

We have thus transformed the problem from optimal Toeplitz-plus-Hankel approximation to optimal Hermitian Toeplitz approximation, which can be easily solved [7]. More specifically, given a $n \times n$ matrix $C = [c_{ij}]$, its optimal Hermitian Toeplitz approximation \tilde{C} can be computed as

$$\tilde{c}_{ii} = \frac{1}{n} \sum_{k=1}^n c_{kk}; \quad \tilde{c}_{i(i+k)} = \frac{1}{2(n-k)} \sum_{m=1}^{n-k} (c_{m(m+k)} + c_{(m+k)m}^*) ; \forall i = 1, \dots, n; k = 1, \dots, n-i \quad (15)$$

where $*$ denotes complex conjugate and $\tilde{c}_{ji} = \tilde{c}_{ij}^*$. After the approximation (Toeplitzation), the resulting Toeplitz-plus-Hankel approximation is

$$U(HT)U^H = T + H \quad (16)$$

The overall procedure to find the optimal symmetric Toeplitz plus skew-persymmetric Hankel matrix \tilde{R} (Toeplitz-plus-Hankelization) to the data covariance matrix R can be summarized as follows:

Given the data covariance matrix R :

1. Perform forward transformation $C = U^H RU$;
2. Perform Hermitian Toeplitzation of $C \rightarrow \tilde{C}$ using (15);
3. Perform inverse transformation $\tilde{R} = U\tilde{C}U^H$

B. Example

$$\text{Let } R = \begin{bmatrix} 2 & 2 & 1 & 5 \\ 1 & 1 & 3 & 6 \\ 1 & 2 & 4 & 3 \\ 2 & 3 & 6 & 8 \end{bmatrix}$$

$$(i) U^H R U = \begin{bmatrix} 5 + 1.5j & 4 - j & 2 - 2j & 3.5 - 3j \\ 2 + 2.5j & 2.5 + 0.5j & 2.5 - 1.5j & 3.5 - j \\ 3.5 + j & 2.5 + 1.5j & 2.5 - 0.5j & 2 - 2.5j \\ 3.5 + 3j & 2 + 2j & 4 + j & 5 - 1.5j \end{bmatrix}$$

$$(ii) \text{ Hermitian Toeplitzation } HT = \begin{bmatrix} 3.75 & 2.83 - 1.67j & 2.75 - 1.5j & 3.5 - 3j \\ 2.83 + 1.67j & 3.75 & 2.83 - 1.67j & 2.75 - 1.5j \\ 2.75 + 1.5j & 2.83 + 1.67j & 3.75 & 2.83 - 1.67j \\ 3.5 + 3j & 2.75 + 1.5j & 2.83 + 1.67j & 3.75 \end{bmatrix}$$

$$(iii) U(HT)U^H = \underbrace{\begin{bmatrix} 3.75 & 2.83 & 2.75 & 3.5 \\ 2.83 & 3.75 & 2.83 & 2.75 \\ 2.75 & 2.83 & 3.75 & 2.83 \\ 3.5 & 2.75 & 2.83 & 3.75 \end{bmatrix}}_{\text{Toeplitz matrix}} + \underbrace{\begin{bmatrix} -3 & -1.5 & -1.67 & 0 \\ -1.5 & -1.67 & 0 & 1.67 \\ -1.67 & 0 & 1.67 & 1.5 \\ 0 & 1.67 & 1.5 & 3 \end{bmatrix}}_{\text{Hankel matrix}} = \begin{bmatrix} 0.75 & 1.33 & 1.08 & 3.5 \\ 1.33 & 2.08 & 2.83 & 4.42 \\ 1.08 & 2.83 & 5.42 & 4.33 \\ 3.5 & 4.42 & 4.33 & 6.75 \end{bmatrix}$$

The Hilbert-Schmidt norm of the error for this Toeplitz-plus-Hankel approximation is equal to 4.6. For Toeplitz-only approximation, the error norm is 8.05. The main reason for using these transformations is that the Hermitian Toeplitzation problem can be easily solved by simply averaging the elements along diagonals. However, to do this, we need four complex matrix multiplications for the forward and inverse transformations. In the next section we show that this procedure reduces to simple averaging operations along the diagonals and antidiagonals.

C. Modified Projection Method

Consider the following $(2n - 1) \times n$ matrices:

$$\begin{bmatrix} 1 & \dots & \dots & 0 \\ \vdots & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 0 & \dots & \dots & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & \dots & 0 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & \dots & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 1 & \dots & \dots & 0 \end{bmatrix}, \begin{bmatrix} 0 & \dots & 1 & 0 \\ \vdots & \vdots & \vdots & -1 \\ 1 & \vdots & \vdots & \vdots \\ 0 & -1 & \dots & 0 \end{bmatrix}, \dots, \begin{bmatrix} 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -1 \end{bmatrix} \quad (17)$$

We now show that the above matrices are mutually orthogonal in the inner product (5), and also span a subspace in which every element can be represented as the sum of a symmetric Toeplitz matrix and a skew-persymmetric Hankel matrix.

Theorem 1 *The $2n - 1$ matrices in (17) are mutually orthogonal, and hence form a set of basis functions for a subspace.*

Proof: see Appendix B.

Theorem 2 *A matrix can be represented as the sum of a symmetric Toeplitz matrix and a skew-persymmetric Hankel matrix if and only if it lies in the subspace spanned by the basis functions in (17).*

Proof: see Appendix C.

From the above theorems, the optimal approximation \tilde{R} of any matrix R by the sum of a symmetric Toeplitz matrix and a skew-persymmetric Hankel matrix can be computed as

$$\tilde{R} = \underbrace{\sum_{i=0}^{n-1} \frac{\langle R, T_i \rangle}{\langle T_i, T_i \rangle} T_i}_{\text{averaging along diagonals}} + \underbrace{\sum_{i=1}^{n-1} \frac{\langle R, H_i \rangle}{\langle H_i, H_i \rangle} H_i}_{\text{averaging along antidiagonals}} \quad (18)$$

Both the modified projection method and the transformation method compute the projection onto the subspace of symmetric Toeplitz plus skew-persymmetric Hankel matrices, as shown by (14). Since this subspace is convex, the projection is unique. Hence both methods are equivalent. This can also be shown by going through the transformation method algebraically, and showing that the result is the modified projection method.

D. Example

Consider the same $R = \begin{bmatrix} 2 & 2 & 1 & 5 \\ 1 & 1 & 3 & 6 \\ 1 & 2 & 4 & 3 \\ 2 & 3 & 6 & 8 \end{bmatrix}$

$$\begin{aligned} \tilde{R} = & \frac{2+1+4+8}{4} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \frac{2+3+3+1+2+6}{6} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} + \frac{1+6+1+3}{4} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ & + \frac{5+2}{2} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \frac{1+1+1-6-4-3}{6} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} + \frac{2+1-3-6}{4} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \end{aligned}$$

$$+ \frac{2-8}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0.75 & 1.33 & 1.08 & 3.5 \\ 1.33 & 2.08 & 2.83 & 4.42 \\ 1.08 & 2.83 & 5.42 & 4.33 \\ 3.5 & 4.42 & 4.33 & 6.75 \end{bmatrix}$$

This example verifies that both the transformation method and the modified projection method produce the same results. However, the latter method only requires averaging along diagonals and antidiagonals, which is much easier than the matrix multiplications.

Incorporation of additional constraints such as rank constraint and positive definiteness has been studied in the Hermitian Toeplitz case [8]. These additional constraints can easily be incorporated into the Toeplitz-plus-Hankel case in transformation method, since they are preserved by the transformation (13). This is why the transformation method was presented separately.

V OPTIMAL SYMMETRIC TOEPLITZ BLOCK-TOEPLITZ PLUS SKEW-PERSYMMETRIC HANKEL BLOCK-HANKEL APPROXIMATION

Block data covariance matrices occur in many multichannel and multidimensional problems [9]. To utilize the fast algorithm developed in [3] for computing the optimal prediction filters, we need to find an optimal block Toeplitz-plus-Hankel approximation to a block data covariance matrix.

A. Multichannel Generalization of Previous Results

We focus on the symmetric Toeplitz block-Toeplitz plus skew-persymmetric Hankel block-Hankel case with $n \times p \times p$ blocks, where n is even. If a matrix \mathbf{R} has such structure, then it can be represented by

$$\mathbf{R} = \underbrace{\begin{bmatrix} \tilde{R}_0 & \dots & \tilde{R}_{-(n-1)} \\ \vdots & \tilde{R}_0 & \vdots \\ k_{n-1} & \dots & \tilde{R}_0 \end{bmatrix}}_{\text{symmetric Toeplitz block-Toeplitz matrix}} + \underbrace{\begin{bmatrix} \tilde{R}_{-(n-1)}J & \dots & \tilde{R}_0J \\ \vdots & \tilde{R}_0J & \vdots \\ \tilde{R}_0J & \dots & \tilde{R}_{n-1}J \end{bmatrix}}_{\text{skew-persymmetric Hankel block-Hankel matrix}} \quad (19)$$

where $\bar{R}_i = \bar{R}_{-i}^T$, $\tilde{R}_i = -\tilde{R}_{-i}^T$, and both \bar{R}_i and \tilde{R}_i are $p \times p$ Toeplitz matrices for $-(n-1) \leq i \leq n-1$. For this type of matrix, we can extend the unitary transform (13) from [6] to the following multichannel case:

$$U = \frac{1}{2} \begin{bmatrix} (1-j)I_p & \dots & 0 & 0 & \dots & (1+j)J_p \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & (1-j)I_p & (1+j)J_p & \dots & 0 \\ 0 & \dots & (1+j)J_p & (1-j)I_p & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (1+j)J_p & \dots & 0 & 0 & \dots & (1-j)I_p \end{bmatrix} \quad (20)$$

and we also have

$$U^* = U^{-1} = \frac{1}{2} \begin{bmatrix} (1+j)I_{\frac{n \times p}{2}} & (1-j)J_{\frac{n \times p}{2}} \\ (1-j)J_{\frac{n \times p}{2}} & (1+j)I_{\frac{n \times p}{2}} \end{bmatrix} \quad (21)$$

Then

$$U^{-1}RU = \begin{bmatrix} R_0 & \dots & R_{-(n-1)} \\ \vdots & R_0 & \vdots \\ R_{n-1} & \dots & R_0 \end{bmatrix} \quad (22)$$

where $R_i = \bar{R}_i + j\tilde{R}_i$. We have $R_i = R_{-i}^H$ by the assumption of $\bar{R}_i = \bar{R}_{-i}^T$, and $\tilde{R}_i = -\tilde{R}_{-i}^T$, so that the block matrix resulting from the transformation is a block Hermitian Toeplitz matrix.

The procedures for computing the optimal symmetric Toeplitz block-Toeplitz plus skew-persymmetric Hankel block-Hankel matrix are the same as before, except now all the matrices become block matrices.

To avoid the matrix multiplications, the projection method is applicable, with some modifications of the basis functions. We can represent the $p \times p$ matrices in (6) and (7) respectively as

$$[\bar{T}_i]_{jk} = \delta_{k-j-i}; [\bar{H}_i]_{jk} = \delta_{k+j-(p+1-i)}, \text{ for all } 1 \leq j, k \leq p, \text{ and } -(p-1) \leq i \leq p-1 \quad (23)$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. The n symmetric Toeplitz matrices and the $n-1$ skew-persymmetric Hankel matrices in (17) can then be respectively represented as

$$[T_0]_{i,j} = \delta_{i-j}; [T_l]_{i,j} = \delta_{i-j-l} + \delta_{j-i-l}, \quad 1 \leq i, j \leq n, \quad l = 1, \dots, n-1 \quad (24)$$

$$[H_l]_{i,j} = \delta_{i+j-(n+1-l)} - \delta_{i+j-(n+1+l)}, \quad 1 \leq i, j \leq n, \quad l = 1, \dots, n-1 \quad (25)$$

Then the basis functions for the subspace of symmetric Toeplitz block-Toeplitz matrices are

$$T_0 \odot \bar{T}_0; T_0 \odot (\bar{T}_l + \bar{T}_{-l}); T_i \odot \bar{T}_j \quad \text{for } l = 1, \dots, p-1; i = 1, \dots, n-1; j = -(p-1), \dots, p-1 \quad (26)$$

and the basis functions for the skew-persymmetric Hankel block-Hankel matrices are

$$(J \cdot T_0) \odot (\bar{H}_l - \bar{H}_{-l}); H_i \odot \bar{H}_j \quad \text{for } l = 1, \dots, p-1; i = 1, \dots, n-1; j = -(p-1), \dots, p-1 \quad (27)$$

where \odot is the modified outer product operation, defined as $A \odot B \triangleq \{a_{m,n}B\}$ if $n \geq m$, and $\{a_{m,n}(JB\bar{J})\}$ if $n < m$, where $a_{m,n}$ is the (m,n) -th element of matrix A . Since all the elements in $T_i(\bar{T}_i)$ and $H_i(\bar{H}_i)$ are either 0 or 1, the outer product is a simple operation in this special case. These two sets of matrices are easily verified to be orthogonal by following the same steps used in Theorem 1. The resulting approximation for a symmetric Toeplitz block-Toeplitz plus skew-persymmetric Hankel block-Hankel matrix is equivalent to projecting this block matrix on the subspace spanned by the basis functions in (26) and (27), which leads to averaging the diagonal and antidiagonal blocks, and then the diagonals and antidiagonals of each block.

B. Example

$$\text{Assume } \mathbf{R} = \begin{bmatrix} 5 & 2 & 4 & 9 & 3 & 1 \\ 8 & 4 & 2 & 7 & 6 & 12 \\ 1 & 9 & 3 & 6 & 2 & 4 \\ 1 & 2 & 3 & 4 & 2 & 8 \\ 2 & 3 & 14 & 6 & 5 & 2 \\ 4 & 7 & 3 & 5 & 6 & 15 \end{bmatrix}$$

Then, after forward transformation, block Hermitian Toeplitzation, and inverse transformation (or using the projection method directly), we obtain the optimal approximation $\tilde{\mathbf{R}} =$

$$\begin{bmatrix} 6 & 4.5 & 5.375 & 4.25 & 6 & 2.5 \\ 4.5 & 6 & 4.75 & 5.375 & 4.5 & 6 \\ 5.375 & 4.75 & 6 & 4.5 & 5.375 & 4.25 \\ 4.25 & 5.375 & 4.5 & 6 & 4.75 & 5.375 \\ 6 & 4.5 & 5.375 & 4.75 & 6 & 4.5 \\ 2.5 & 6 & 4.25 & 5.375 & 4.5 & 6 \end{bmatrix} + \begin{bmatrix} -5 & 0.5 & -2 & 0.75 & -2.5 & 0 \\ 0.5 & -0.5 & 0.75 & -1.75 & 0 & 2.5 \\ -2 & 0.75 & -2 & 0 & 1.75 & -0.75 \\ 0.75 & -1.75 & 0 & 2 & -0.75 & 2 \\ -2.5 & 0 & 1.75 & -0.75 & 0.5 & -0.5 \\ 0 & 2.5 & -0.75 & 2 & -0.5 & 5 \end{bmatrix}$$

The fast algorithm of [3] was designed for linear prediction on a polar raster. Since covariance functions on a polar raster are periodic in the angular variables, the associated covariance matrices

will have circulant blocks. The basis (26)-(27) should be modified to

$$T_i \odot \bar{T}_0; T_i \odot (\bar{T}_j + \bar{T}_{-(p-j)}) \quad \text{for } i = 0, \dots, n-1; j = 1, \dots, p-1 \quad (28)$$

for the Toeplitz block-circulant matrices and

$$H_i \odot (J \cdot \bar{T}_0); H_i \odot (\bar{H}_j + \bar{H}_{-(p-j)}) \quad \text{for } i = 1, \dots, n-1; j = 1, \dots, p-1 \quad (29)$$

for the Hankel block-circulant matrices. These basis functions are easily shown to be orthogonal, so the projection on this basis can again be found by averaging along diagonals and antidiagonals.

VI CONCLUSION

In this paper, the well-known problem of "Toeplitzation" of a data covariance matrix has been extended to Toeplitz-plus-Hankel approximation of matrices. The general solution can be computed by projecting the given data covariance matrix on the space of Toeplitz-plus-Hankel matrices. Although the basis functions for this subspace can be recursively generated, as the size of the matrix grows large, the Gram-Schmidt orthogonalization requires much computation. To obtain a simpler algorithm, we can restrict ourselves to the subspace of symmetric Toeplitz plus skew-persymmetric Hankel matrices, for which the optimal approximation can be efficiently computed by averaging along diagonals and antidiagonals. We also show that the same result can be achieved by a unitary transformation along with Hermitian Toeplitzation; the latter algorithm permits additional constraints such as rank constraints and positive definiteness.

For the multichannel and multidimensional problems, approximation for a block data covariance matrix is also considered. The optimal symmetric Toeplitz block-Toeplitz plus skew-persymmetric Hankel block-Hankel matrix can be derived either by using the unitary transformation along with block Hermitian Toeplitzation, or the more efficient projection method.

ACKNOWLEDGMENT

This work was supported by the Air Force Office of Scientific Research under grant # AFOSR-89-0017.

APPENDIX A

Prove that there are only $4n - 4$ matrices left in (6) and (7) after the Gram-Schmidt orthogonalization.

Since the matrices in (6) and (7) span the space of Toeplitz-plus-Hankel matrices, the number of orthogonal matrices in this subspace is equal to that of the linearly independent matrices in (6) and (7).

We use the same notation as in (23) for \bar{T}_i and \bar{H}_i , with the replacement of p by n . Since $\{\bar{T}_i\}_{i=-(n-1)}^{n-1}$ is a set of linearly independent (also orthogonal) matrices, we adjoin $\{\bar{T}_i\}_{i=-(n-1)}^{n-1}$ with the elements in $\{\bar{H}_i\}_{i=-(n-1)}^{n-1}$ in the following order : $\bar{H}_0, \bar{H}_1, \bar{H}_{-1}, \dots, \bar{H}_{n-1}, \bar{H}_{-(2n-1)}$. If the newly added element is linearly dependent on the previous matrices, then we remove it. So the number of matrices remaining form a set of linearly independent matrices.

If a matrix is linearly dependent with a set of matrices, then we must be able to find a sequence of lines such that each non-zero element in these dependent matrices is crossed by these lines at least twice. Note that the non-zero elements in matrices of (6) and (7) are some specific lines in either NE-SW or NW-SE directions. It is easy to check that if the index j is even (odd), then for the above condition to hold the elements $\bar{H}_{\pm(n-1)}$ ($\bar{H}_{\pm(n-2)}$) are always required.

The only other possibilities are $\bar{H}_{-(n-2)}$ and \bar{H}_{n-1} . If n is even, we have

$$\sum_{i \text{ even}} \bar{H}_i = \sum_{i \text{ odd}} \bar{T}_i ; \quad \sum_{\forall i} \bar{H}_i = \sum_{\forall i} \bar{T}_i \quad (30)$$

Reordering the terms, we get

$$\bar{H}_{-(n-2)} = \sum_{i \text{ odd}} \bar{T}_i - \sum_{i \text{ even}, i \neq -(n-2)} \bar{H}_i ; \quad \bar{H}_{-(n-1)} = \sum_{\forall i} \bar{T}_i - \sum_{i \neq (n-1)} \bar{H}_i \quad (31)$$

which means that $\bar{H}_{-(n-2)}$ and $\bar{H}_{-(n-1)}$ are linearly dependent on the other matrices. If n is odd, simply interchange "even" and "odd" in the above argument. Therefore, there are $2(2n - 1) - 2 = 4n - 4$ linearly independent matrices in (6) and (7). ■

APPENDIX B

Proof of Theorem 1

We use the same notation as in (24) and (25) for T_i and H_i . It is easy to verify that $\langle T_m, T_l \rangle = 0$ and $\langle H_m, H_l \rangle = 0$ in the sense of (5) if $m \neq l$ (see [5]). Therefore, we only need to show that T_m and H_l are mutually orthogonal. From (24) and (25), we have ($m \neq 0$)

$$\begin{aligned}
 \langle T_m, H_l \rangle &= \text{Trace}[T_m^T H_l] = \sum_{i=1}^n \sum_{k=1}^n [T_m]_{i,k} [H_l]_{k,i} \\
 &= \sum_{i=1}^n \sum_{k=1}^n [\delta_{i-k-m} + \delta_{k-i-m}] [\delta_{k+i-(n+1-l)} - \delta_{k+i-(n+1+l)}] \\
 &= \sum_{i=1}^n \sum_{k=1}^n \underbrace{-\delta_{i-k-m} \delta_{k+i-(n+1+l)}}_{\text{Point A}} + \underbrace{\delta_{i-k-m} \delta_{k+i-(n+1-l)}}_{\text{Point B}} - \underbrace{\delta_{k-i-m} \delta_{k+i-(n+1+l)}}_{\text{Point C}} + \underbrace{\delta_{k-i-m} \delta_{k+i-(n+1-l)}}_{\text{Point D}} \quad (32)
 \end{aligned}$$

So the solution of (32) can then be determined by the intersections of these four straight lines, i.e., $i - k = m$, $k - i = m$, $k + i = n + 1 + l$, and $k + i = n + 1 - l$, as shown in Figure 1. If $m \neq 0$, by symmetry these four lines either do not intersect at all ($A = B = C = D = 0$), or have four intersections, for which $A = B = C = D = 1$. In both cases, (32) is equal to zero. If $m = 0$, then there are always two intersections between the lines $k + i = n + 1 + l$, $k + i = n + 1 - l$, and $i = k$, and the result is still equal to zero. Therefore, T_m and H_l are mutually orthogonal, and these $2n - 1$ matrices form a set of basis functions. ■

APPENDIX C

Proof of Theorem 2

(a) "If" part: Any vector in this subspace can be represented as $\sum_{i=0}^{n-1} a_i T_i + \sum_{i=1}^{n-1} a'_i H_i$. It is obvious that the first sum is a symmetric Toeplitz matrix, and the second sum is a skew-persymmetric Hankel matrix. (b) "Only if" part: If a matrix C can be represented as the sum of symmetric Toeplitz matrix \tilde{T} and skew-persymmetric Hankel matrix \tilde{H} , then C can be represented as $C = \sum_{i=1}^n [\tilde{T}]_{ii} T_{i-1} + \sum_{i=1}^{n-1} [\tilde{H}]_{1(n-i)} H_i$. Hence this matrix lies in the space spanned by the basis functions of (17). ■

References

- [1] A. E. Yagle, *New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices*, to appear in IEEE Trans. SP, Nov. 1991.
- [2] W.-H. Fang, A.E. Yagle, *Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster*, Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc., Albuquerque, NM, April 1990, pp. 2017-2020.
- [3] W.-H. Fang, A.E. Yagle, *Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster*, submitted to IEEE Trans. SP, Nov. 1989.
- [4] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [5] H. Wang, G. Wakefield, *Signal-Subspace Approximation for Line Spectrum Estimation*, Proc. Int'l. Conf. on Acoust., Speech, Sig. Proc., Dallas, TX, April 1987, pp. 2054-2057.
- [6] F. Noor, S. D. Morgera *A Hermitian Toeplitz Matrix is Unitarily Similar to a Real Toeplitz-plus-Hankel Matrix*, preprint, Dept. of Elec. Engin., McGill University, Montreal, P.Q. Canada.
- [7] H. Wang, *Eigenstructure-Based Performance Analysis and Toeplitz Approximation for Direction-of-Arrival Estimation*, Ph.D. Thesis, Dept. of EECS, University of Michigan, Ann Arbor, MI, 1990.
- [8] J. A. Cadzow, *Signal Enhancement-A Composite Property Mapping Algorithm*, IEEE Trans. ASSP, vol. ASSP-36, no.1, pp. 49-62, Jan. 1988
- [9] S. M. Kay, *Modern Spectral Estimation*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

FIGURE HEADING

1. Figure 1 : The intersections of $i - k = m$, $k - i = m$, $k + i = n + 1 + l$, and $k + i = n + 1 - l$.

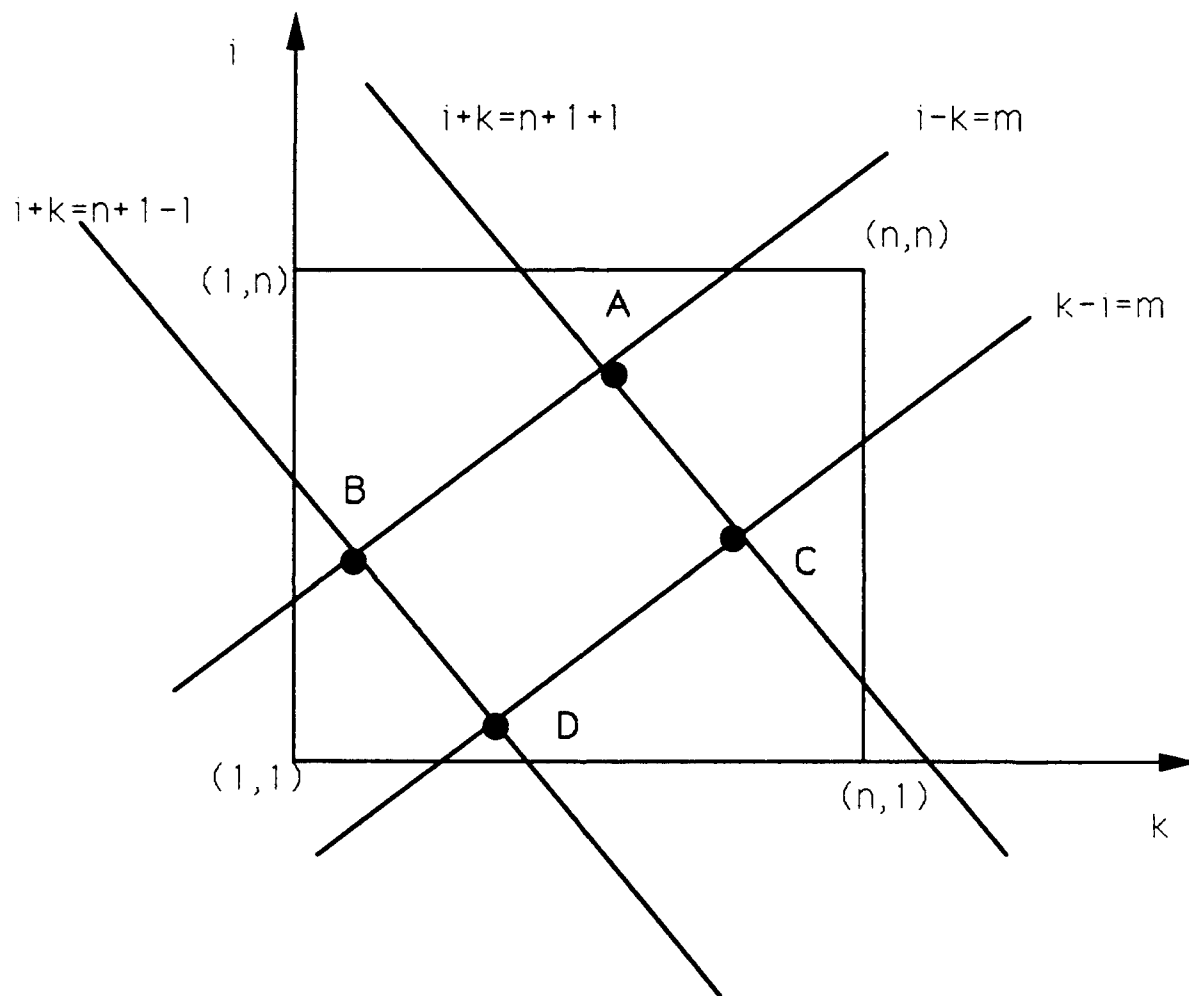


Figure 1:

APPENDIX G

W.-H. Fang and A.E. Yagle, "Two-Dimensional Linear Prediction and Spectral Estimation on a Polar Raster," submitted to *IEEE Trans. Signal Processing*.

Two-Dimensional Linear Prediction and Spectral Estimation on a Polar Raster

Wen-Hsien Fang and Andrew E. Yagle
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-2122

June 1991

Abstract

A zero-mean homogeneous random field is defined on a discrete polar raster. Given sample values inside a disk of finite radius, we wish to estimate the field's power spectral density using linear prediction. Issues arising here include estimation of covariance lags, and extendibility of a finite set of lag estimates into a positive semi-definite covariance extension (required for a meaningful spectral density). We give a generalized autocorrelation procedure that guarantees positive semi-definite covariance estimates. It first interpolates the data using Gaussians, computes its Radon transform, and applies familiar one-dimensional techniques to each slice. Some numerical examples are provided to justify the validity of the proposed procedure. We also propose a correlation matching covariance extension procedure that uses the Radon transform to extend a given set of covariance lags to the entire plane, when this is possible, and discuss circumstances for which this is impossible.

I INTRODUCTION

In many applications, such as tomographic imaging problems solved by filtered back-projection [1], and spotlight synthetic aperture radar [2], data are collected on a polar raster of points, rather than on a rectangular lattice. To process such data, e.g. remove undesired frequency components; we need to estimate the power spectral density for data defined on a polar raster.

The obvious approach of simply estimating the 1-D power spectral density independently along each slice will give an incorrect answer, since the 2-D Fourier transform on a polar raster is not given by the 1-D Fourier transform along each slice. One approach, the 2-D periodogram, is to interpolate the data onto a rectangular lattice, and then take the 2-D Fourier transform of the resampled values. We note here that for a rectangular raster, 1-D spectral estimation techniques have been applied, first by columns, then by rows, in some "separable" 2-D spectral estimators [3, 4]. While these separable estimators do compute the 2-D Fourier transform correctly, they neglect correlation between rows and columns.

A major problem with the 2-D periodogram is the poor resolution of spectral estimates based on a small amount of data [5]. This is due to truncation of the covariance lags, since only a finite amount of data samples is available. To overcome this difficulty in 1-D, parametric modeling is used to extend the finite set of covariance lags. Linear prediction (AR modeling) is the most common approach due to its simplicity and high-resolution spectral estimates. New contributions of this paper include the following:

1. An "autocorrelation" 2-D spectrum estimation procedure which uses the Radon transform to transform the 2-D problem into an uncorrelated set of 1-D spectrum estimation problems. It is an autocorrelation method in that all unknown values are windowed to zero, as in the autocorrelation method for 1-D linear prediction, for computing the Radon transform. It differs from a previous Radon-based 2-D spectrum estimation procedure [6] in the following three ways:

- (a) The Radon transform is computed in a different manner that ensures a non-negative esti-

mate of power spectral density;

(b) The use of 1-D linear prediction to obtain finer-resolution 1-D spectrum estimates along each 2-D slice;

(c) Discussion of the effects of the 1-D covariance extension along each slice on the 2-D covariance (viz. correlation matching holds in the Radon transform domain, but not in the 2-D domain);

2. A new 2-D covariance extension procedure that extends a set of 2-D covariance lags defined in a finite disk to the entire plane, when this is possible. Unlike the first procedure, this procedure has the correlation matching property of preserving the given covariance lags in the 2-D domain;
3. A discussion of various interpolating functions used to compute the discrete Radon transform, and implications of their use for 2-D spectrum estimation.

A. Review of 2-D Linear Prediction on a Rectangular Raster

Many aspects of 1-D linear prediction have been shown to generalize to the 2-D case defined on a rectangular raster [7]. For example, stability and minimum phase properties are still related to reflection coefficients [7]. However, two vital aspects do not generalize to the 2-D case:

1. *Causality*, which has a clear definition in the 1-D case, has been defined in at least two different ways on a 2-D rectangular raster. *Asymmetric half-plane* causality [7] splits the 2-D raster into "past" and "future" half-planes; the 2-D AR model has support in the "past". *Quarter-plane* causality [8] means that the 2-D autoregressive (AR) model has support in a quarter-plane, e.g. to the "southwest" of the present point. Since quarter-plane causality is a special case of asymmetric half-plane causality, we consider only the latter in the sequel.
2. An essential feature of 1-D linear prediction is *covariance extendibility*: Given a finite positive semi-definite (psd) set of covariance lag estimates, it is always possible to extend this set into an

infinite psd set of covariance lags. This is important since a non-psd set of covariance lags will lead to negative values in the estimated power spectral density. However, this property does not extend to the 2-D case on a rectangular raster.

For asymmetric half-plane causality, the region of support for the 2-D AR model is infinite, so that truncation is clearly necessary. This truncation is the cause of much of the difficulty in 2-D linear prediction; it results in a *discontinuous* region of support, and even in the 1-D case a discontinuous region of support creates problems. In ([7], p. 59) a 1-D example with discontinuous support results in a non-minimum phase AR filter that does not satisfy the correlation-matching property. Indeed, a finite set of 2-D psd covariances with discontinuous support may not even have a psd extension over the entire plane [9].

The cause of the difficulties can be seen by examining the Yule-Walker equations for determining the AR filter coefficients from the covariance lag estimates. These equations have block-Toeplitz form, so that the number of covariance lag estimates exceeds the number of AR filter coefficients (see [5], p. 495 for a specific example). This has two implications:

1. An infinite number of different covariance lag estimates can be associated with the same AR model. Hence the correlation matching property, which guarantees that the spectral estimate will be consistent with the finite set of lag estimates, no longer holds:
2. Covariance extension from a finite set of estimated lags requires recursion using the 2-D AR model, over an asymmetric half-plane. Since the region of support is infinite, and only a finite set of lags estimates are given, truncation is necessary, and this may result in a non-psd covariance extension.

B. 2-D Linear Prediction on a Polar Raster

In this paper we address, for the first time, similar questions for a random field defined on a *polar raster*. On a polar raster, causality is defined unambiguously in terms of increasing radius; the region

of support for an AR model at any point on a given circle is the disk *inside* the circle. Since this disk is a continuous region of support, the result of [9] is inapplicable.

Indeed, we give an explicit procedure which inputs discrete sample values inside a finite disk, and outputs a set of psd covariance lags. We call this covariance extension, although strictly speaking we are not extending a set of covariance lags, but creating an extended set of psd lags from a finite set of data. In Section VI we propose another algorithm that explicitly extends a finite set of 2-D lags to the entire plane, provided this is possible.

In this paper, we propose using the Radon transform to decouple the 2-D spectral estimation problem into a set of 1-D problems. The projection-slice theorem tells us that there are two ways to compute the 2-D Fourier transform: (1) we can either compute it directly by taking the 2-D Fourier transform; or (2) we can take the Radon transform first, and then apply the 1-D Fourier transform along each direction in the spectral domain. This suggests the following algorithm for 2-D spectral estimation: (1) take the Radon transform of the data; (2) extrapolate the 1-D covariance lags in the Radon transform domain along each direction, using 1-D linear prediction; and then (3) superposing the 1-D spectral estimates to form a 2-D spectral estimate, defined on a polar raster.

Note that the available data are *discrete* samples, but the projection-slice theorem only holds for *continuous* data, so we need to find some *interpolating functions* to interpolate the discrete data. Since the sampling theorem on a polar raster is very different from that on the rectangular lattice, the interpolating functions for a band-limited signal are quite complicated [10, 11]. In this paper, we propose using gaussian interpolating functions to compute the Radon transform of the given discrete data. A complete discussion of the merits of gaussian vs. other interpolating function is also addressed. It should be noted that our proposed “interpolating” function does not agree with the original specified discrete data points; indeed it should more properly be termed a “defocusing” function. To make it easier for the reader, we give “interpolating function” a definition slightly different from the usual; see Section III.

This paper is organized as follows. Section II proposes a psd covariance extension method using the Radon transform. Section III discusses the choice of interpolating functions to transform the discrete data samples into continuous data. The analytically explicit procedure using the gaussian interpolating functions is then given in Section IV. This procedure can be used to provide a high-resolution spectral estimate for points defined on a polar raster. Some numerical examples are given in Section V. In Section VI, we propose a 2-D psd covariance extension technique that also has the correlation matching property, provided that a 2-D psd extension exists. Section VII concludes with a summary.

II 2-D LINEAR PREDICTION AND PSD COVARIANCE EXTENSION ON A POLAR RASTER

A. Problem Formulation

The problem considered is as follows. A set of data is defined on a polar raster. We are given discrete sample values $\{f(i, m), 0 \leq i \leq N, 1 \leq m \leq M\}$ at the points $(i\delta_r, 2\pi m/M)$ on the polar raster, as shown in Figure 1; i is integer radius from the origin, δ_r is the radial spacing, and m is the integer index of angular position, corresponding to an angle of $2\pi m/M$ radians. The goal is to compute a psd set of covariance lags everywhere in the plane.

The assumption of discrete samples is required, since any numerical procedure will ultimately require discretization. We point out here that if the data is generated from an isotropic random field which is bandlimited in wavenumber to a disk of radius π , and $M > 2\pi N$, then the discrete sampled points $\{f(i, m), 0 \leq i \leq N, 1 \leq m \leq M\}$ may be interpolated to give the exact value of the random field everywhere in the disk of radius N [10].

B. The 2-D Radon Transform and Projection-Slice Theorem

In order to decouple the 2-D linear prediction problem into a set of 1-D linear prediction problems along each slice, it is necessary to first compute the *Radon transform* of the data. The 2-D Radon

transform is defined as

$$\hat{f}(t, \theta) = \mathcal{R}\{f(x, y)\} = \int \int f(x, y) \delta(t - x \cos \theta - y \sin \theta) dx dy \quad (1)$$

so that the Radon transform is the set of projections or line integrals of $f(x, y)$ along all possible lines.

An important property of the Radon transform is the *projection-slice theorem*, which states that the 2-D Fourier transform $F(k, \theta)$ in polar coordinates of $f(x, y)$ can be computed by taking 1-D Fourier transforms along each slice of the *Radon transform* of $f(x, y)$ so that [12]

$$F(k, \theta) = \mathcal{FF}\{f(x, y)\} = \mathcal{F}_{t \rightarrow k}\{\hat{f}(t, \theta)\} \quad (2)$$

where $\mathcal{F}_{t \rightarrow k}$ denotes a 1-D Fourier transform taking t into wavenumber k , and $\hat{f}(t, \theta)$ is computed using (1). A discrete version of the projection-slice theorem has been used to develop a fast algorithm for computing 2-D discrete Fourier transforms: first the discrete Radon transform is computed, and then 1-D discrete Fourier transforms are computed along each slice of the Radon transform [13]. Since both transforms are parallelizable, this can save computation time.

For a homogeneous random field, it may be shown that the Radon transform is a *whitening* transform: each slice of the Radon transform of a homogeneous random field is uncorrelated with each other slice [14]. This suggests that the 2-D linear prediction problem can be decoupled into a set of independent 1-D linear prediction problems by Radon transforming the data. This approach was taken in [6]; however, [6] did not consider the problems of linear prediction on a polar raster, from a finite disk of data, correlation matching, and psd covariance extension.

C. Procedure for 2-D Covariance Extension

Clearly computation of the Radon transform from the data will require interpolation. In the next section, we will discuss how to choose an interpolating function to transform data from the discrete domain into the continuous domain. At present, for convenience we assume that the data are continuous and inside a disk of finite radius.

Since we have data only inside a disk of finite radius, we propose an "autocorrelation" method in which the unknown data are windowed to zero for purposes of computing the Radon transform. The Radon transform is then computed analytically. Finally, the 1-D autocorrelation form of linear prediction is used on each slice of the Radon transform to get a set of psd covariance estimates.

The term "autocorrelation method", in the linear prediction sense of the term, is justified due to the following two properties of the Radon transform:

1. Let $\hat{x}(t, \phi)$ be the Radon transform of the random field $x(r, \theta)$ (using polar coordinates throughout). Note from (1) that for any $T > 0$ $\{\hat{x}(t, \phi), t > T\}$ depends only on $\{x(r, \theta), r > T\}$. Hence windowing the data to zero for $r > N$ is equivalent to windowing its Radon transform to zero for $t > N$;
2. Using (2), it is clear that

$$\mathcal{R}\{f(x, y) ** g(x, y)\} = \hat{f}(t, \theta) * \hat{g}(t, \theta)$$

where $**$ denotes 2-D convolution and $*$ denotes 1-D convolution in t . Setting $f(x, y) = x(r, \theta)$ and $g(x, y) = x(r, -\theta)$ in (2) shows that the following two methods are equivalent:

- (a) Windowing the Radon transform of the data to zero, and then forming the covariance lag estimates from these Radon transforms;
- (b) Forming the covariance lag estimates directly from the windowed data (the autocorrelation method of linear prediction), and then Radon transforming the lag estimates.

III COMPUTATION OF DISCRETE RADON TRANSFORM

In this section we discuss the computation of the Radon transform of a function defined on a discrete lattice of points. We call such a transform a *discrete Radon transform*. The discrete Radon transform will be used in the spectral estimation technique proposed below. To facilitate comparison of our method with various other definitions of the discrete Radon transform, we consider first a

rectangular lattice of discrete points, and then a polar raster of discrete points (the latter is the actual case of interest).

A. Rectangular Lattice

Consider a function $f_{i,j}$ defined on a rectangular lattice of points (i, j) , where i and j are integers such that $-M \leq i, j \leq M$ for some M . Our goal is to define and compute the *discrete Radon transform* of $f_{i,j}$ such that the following properties hold:

1. Computation of the discrete Radon transform requires as little time and storage as possible;
2. The Radon transform possesses the projection-slice property;
3. The Radon transform $\hat{f}(t, \theta)$ of a psd discrete 2-D function $f_{i,j}$ is psd in t for each θ .

Note that ease of invertibility of the discrete Radon transform is not an issue here, since the projection-slice theorem states that the 2-D spectrum on a polar raster is immediately determined from the 1-D spectra of the Radon transforms. Hence ease of computation of the *forward* transform, not the *inverse* transform, is significant.

Our approach is to interpolate $f_{i,j}$ into a continuous function $f(x, y)$, defined as

$$f(x, y) = \sum_{i=-M}^M \sum_{j=-M}^M f_{i,j} \phi(x - i, y - j) \quad (3)$$

where $\phi(x, y)$ is defined here as an *interpolating function*. The discrete Radon transform $\hat{f}(t, \theta)$ of $f_{i,j}$ is then defined to be the same as the Radon transform of its interpolation $f(x, y)$, which is

$$\begin{aligned} \hat{f}(t, \theta) &= \mathcal{R}\{f(x, y)\} = \sum_{i=-M}^M \sum_{j=-M}^M f_{i,j} \mathcal{R}\{\phi(x - i, y - j)\} \\ &= \sum_{i=-M}^M \sum_{j=-M}^M f_{i,j} \hat{\phi}(t - i \cos \theta - j \sin \theta, \theta) \end{aligned} \quad (4)$$

where $\mathcal{R}\{\phi(x, y)\} = \hat{\phi}(t, \theta)$. This definition clearly possesses the projection-slice property. To follow, we consider some common interpolating functions, for more other interpolating functions, see [15].

Some choices of interpolating function $\phi(x, y)$, and the resulting discrete Radon transforms, are:

1. Impulses

Choosing for the interpolating function the 2-D impulse function $\phi(x, y) = \delta(x)\delta(y)$ results in

$$\hat{f}(t, \theta) = \sum_{i=-M}^M \sum_{j=-M}^M f_{i,j} \delta(t - i \cos \theta - j \sin \theta) \quad (5)$$

since $\mathcal{R}\{\delta(x)\delta(y)\} = \delta(t)$.

For this choice of interpolating function, the discrete Radon transform is zero unless the line passes precisely through a lattice point; hence $\hat{f}(t, \theta)$ is zero except for a finite set of t and θ (excluding values found from only a single lattice point). This makes this choice unsuitable for 2-D spectral estimation.

This is the discrete Radon transform defined by Beylkin in [16]. Note that on an infinite 2-D lattice of integers ($M \rightarrow \infty$), the set of lines through the origin for which the discrete Radon transform is non-zero is precisely the set of lines with rational slopes.

2. Square Pixels

A common method of computing the Radon transform of a sampled function is to assume that $f_{i,j}$ represents the value of the square 1×1 pixel centered at coordinates $(x, y) = (i, j)$. The Radon transform is then computed as follows. For each line, multiply the length of the line within a pixel by the value $f_{i,j}$ of that pixel, and sum over all pixels through which that line passes. This method was used in [6] to compute the Radon transform for 2-D spectral estimation.

This pixel assumption is clearly equivalent to using for the interpolating function $\phi(x, y) = \text{rect}(x)\text{rect}(y)$, where $\text{rect}(x) = 1$ if $-1/2 < x < 1/2$, and $= 0$ otherwise. The resulting discrete Radon transform is then

$$\hat{f}(t, \theta) = \sum_{i=-M}^M \sum_{j=-M}^M f_{i,j} \hat{\phi}(t - i \cos \theta - j \sin \theta, \theta)$$

where ([12], p. 62)

$$\hat{\phi}(t, \theta) = \mathcal{R}\{rect(x)rect(y)\} = \begin{cases} \frac{\cos \theta + \sin \theta + 2t}{2 \sin \theta \cos \theta}, & \text{if } -\sin \theta - \cos \theta < 2t < \sin \theta - \cos \theta \\ \frac{1}{\cos \theta}, & \text{if } \sin \theta - \cos \theta < 2t < \cos \theta - \sin \theta \\ \frac{\cos \theta + \sin \theta - 2t}{2 \sin \theta \cos \theta}, & \text{if } \cos \theta - \sin \theta < 2t < \cos \theta + \sin \theta \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

It is clear that this requires a considerable amount of computation, in violation of condition #1 above. It should be noted that the value of this definition of discrete Radon transform is that its inverse Radon transform may be computed by solving a linear (but large) system of equations. However, this is not valuable to us in the context of 2-D spectral estimation, since the 2-D spectrum can be found from the 1-D spectrum immediately using the projection-slice property. Hence there is no reason to make the choice of interpolating functions implicitly made in [6]. A more serious problem is that there is no guarantee that the resulting $\hat{f}(t, \theta)$ will be psd, in violation of condition #3.

3. Sinc Functions

Regarding $f_{i,j}$ as samples of a continuous function bandlimited in *radial* wavenumber to $[-\pi, \pi]$ (note that this may or may not actually be the case), the choice $\phi(x, y) = \frac{0.5J_1(\pi\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}$ leads to

$$\hat{f}(t, \theta) = \sum_{i=-M}^M \sum_{j=-M}^M f_{i,j} \text{sinc}(t - i \cos \theta - j \sin \theta) \quad (7)$$

since $\mathcal{R}\left\{\frac{0.5J_1(\pi\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}}\right\} = \text{sinc}(t)$.

This discrete Radon transform is easily computed, satisfying condition #1. The projection-slice property (condition #2) is automatically satisfied. Condition #3 that $\hat{f}(t, \theta)$ be psd in t may not seem at first glance to be satisfied, but if $f_{i,j}$ is psd, and *regarded as samples of a bandlimited function sampled above the Nyquist rate*, then its interpolation $f(x, y)$ is also psd. This means that the 2-D Fourier transform of $f(x, y)$ is non-negative, and by the projection-slice property the 1-D Fourier transform of $\hat{f}(t, \theta)$ is non-negative for each θ , so that $\hat{f}(t, \theta)$ is psd, as required.

$\hat{f}(t, \theta)$ can also be seen to be psd as follows. First, consider the $j = 0$ terms. These can be interpreted as the interpolation of sampled values $f_{i,0}$ using a discretization length $\Delta_x = \cos \theta < 1$

(corresponding to a sampling rate above the Nyquist rate). Repeating this argument for each value of j , $\hat{f}(t, \theta)$ can be interpreted as a sum of delayed signals, each of which is bandlimited and psd. Hence these sum must be psd. Furthermore, the projection-slice property also implies that $\hat{f}(t, \theta)$ is bandlimited to $[-\pi, \pi]$, so that it may be sampled in t and standard discrete-time 1-D spectral estimation techniques applied to it.

Note that regarding $f_{i,j}$ as samples of a continuous function bandlimited in wavenumber to $-\pi < k_x, k_y < \pi$ leads to the choice $\phi(x, y) = \text{sinc}(x)\text{sinc}(y)$. The lack of radial symmetry in $\phi(x, y)$ makes its Radon transform $\hat{\phi}(t, \theta)$ a θ -dependent *sinc* function. The above argument for (7) is also applicable to this case.

B. Polar Raster

We now consider the same problem, but on a discrete polar raster of points having radius N and M radial slices. This is the problem of interest, since our data is given on such a discrete lattice.

The major difference between the rectangular and polar rasters is that, on a polar raster, translation must be described in terms of polar coordinates. Hence $f_{i,j}$ becomes $f_{i,n}$, where integer i denotes radius and integer n denotes an angle $2\pi n/M$ radians from the horizontal (i.e., the n^{th} radial slice). Equation (3) for interpolating the $f_{i,j}$ must be modified to

$$f(r, \zeta) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \delta[(r, \zeta) - (i, 2\pi n/M)] ** \phi(r, \zeta) \quad (8)$$

where $**$ denotes a 2-D convolution in polar coordinates and

$$\delta[(r, \zeta) - (i, 2\pi n/M)] = \delta(r \cos \zeta - i \cos(2\pi n/M)) \delta(r \sin \zeta - i \sin(2\pi n/M)) \quad (9)$$

is a 2-D impulse.

The discrete Radon transform $\hat{f}(t, \theta)$ of $f_{i,n}$ is again defined to be the same as the Radon transform of its interpolation $f(r, \zeta)$. Using the property that the Radon transform of a convolution is the convolution (in t) of the Radon transforms (obvious from the projection-slice property), equation (8)

is modified to

$$\begin{aligned}\hat{f}(t, \theta) &= \mathcal{R}\{f(r, \zeta)\} = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \mathcal{R}\{\delta[(r, \zeta) - (i, 2\pi n/M)]\} * \mathcal{R}\{\phi(r, \zeta)\} \\ &= \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \delta(t - i \cos(\theta - 2\pi n/M)) * \hat{\phi}(t, \theta) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \hat{\phi}(t - i \cos(\theta - 2\pi n/M), \theta)\end{aligned}$$

where $\mathcal{R}\{\phi(r, \zeta)\} = \hat{\phi}(t, \theta)$ and we have used the fact that $\mathcal{R}\{f(\mathbf{x} - \mathbf{a})\} = \hat{f}(t - \mathbf{e} \cdot \mathbf{a}, \mathbf{e})$ [12], where \mathbf{x} and \mathbf{a} are vectors and \mathbf{e} is a unit vector.

1. Impulses

Choosing for the interpolating function the 2-D impulse function $\phi(r, \zeta) = \delta(r)$ results in

$$\hat{f}(t, \theta) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \delta(t - i \cos(\theta - 2\pi n/M)) \quad (10)$$

For this choice of interpolating function, the discrete Radon transform is again zero unless the line passes precisely through a lattice point. This happens when $t = x \cos \theta + y \sin \theta = i \cos(\theta - 2\pi n/M)$, i.e., $x = i \cos \theta$ and $y = i \sin \theta$. Again, only a finite number of lines pass through more than one lattice point; hence this choice is unsuitable for 2-D spectral estimation.

2. Sinc Functions

The choice $\phi(r, \zeta) = \frac{0.5J_1(\pi r)}{r}$ results in

$$\hat{f}(t, \theta) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \text{sinc}(t - i \cos(\theta - 2\pi n/M)) \quad (11)$$

This is a plausible choice. However, this choice of interpolating function *does NOT correspond to interpolating samples of a bandlimited function*, since the sampling is performed on a polar raster. The problem of interpolating a bandlimited function from samples on a polar raster has been considered in [11]; however [11] required that the samples be taken at non-uniform radial distances, corresponding to the interlaced zeros of Bessel functions of the first kind of various orders. Hence the results of [11] are not applicable here.

3. Gaussian Functions

The choice $\phi(r, \zeta) = e^{-r^2}$ results in

$$\hat{f}(t, \theta) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \sqrt{\pi} e^{-(t-i \cos(\theta-2\pi n/M))^2} \quad (12)$$

since $\mathcal{R}\{e^{-r^2}\} = \sqrt{\pi} e^{-t^2}$. This is easy to compute, satisfying condition #1, and the projection-slice property (condition #2) holds automatically. However, unlike the sinc interpolating functions, a set of psd $f_{i,n}$ guarantees that $\hat{f}(t, \theta)$ will be psd in t , so that condition #3 is also satisfied. This is true since: (1) the Fourier transform of a Gaussian function is also Gaussian; and (2) a Gaussian function is always positive. We now prove that condition #3 is satisfied.

Recall that the interpolated function $f(\mathbf{x})$, where \mathbf{x} is a vector, is defined by

$$f(\mathbf{x}) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} \delta(\mathbf{x} - (i, 2\pi n/M)) ** \phi(\mathbf{x}) \quad (13)$$

where $(i, 2\pi n/M)$ is a point on the polar raster and $**$ denotes a 2-D convolution. Taking the 2-D Fourier transform of this yields

$$F(\mathbf{k}) = \sum_{i=0}^N \sum_{n=1}^M f_{i,n} e^{-j\mathbf{k} \cdot (i, 2\pi n/M)} \Phi(\mathbf{k}) \quad (14)$$

where \mathbf{k} is a wavenumber vector and $\Phi(\mathbf{k}) = \mathcal{FF}\{\phi(\mathbf{x})\}$. We recognize the expression multiplying $\Phi(\mathbf{k})$ as the 2-D discrete-time Fourier transform (2DDTFT) of $f_{i,n}$ in discrete polar coordinates; since $f_{i,n}$ is psd this is non-negative. If $\Phi(\mathbf{k})$ is non-negative, $F(\mathbf{k})$ is also non-negative, and by the projection-slice property $\hat{f}(t, \theta)$ is psd in t . Hence conditions #1-#3 are all satisfied if: (1) $\Phi(\mathbf{k}) > 0$; (2) both $\phi(\mathbf{x})$ and $\hat{\phi}(t, \theta)$ have simple forms *in polar coordinates*; and (3) both $\phi(\mathbf{x})$ and $\hat{\phi}(t, \theta)$ have “reasonable” forms that interpolate the data (this excludes impulses).

C. Choice of Interpolating Function

At one extreme we have the impulse interpolating function, and at the other extreme we have the sinc interpolating function. The gaussian interpolating function occupies a middle ground. Although there is no firm basis for choice, we have chosen the gaussian interpolating function because it occupies the middle ground.

Another reason to choose the gaussian interpolating function is that by varying the variance, we can control the width of the interpolating function in both space and wavenumber. Note from (14) that the interpolation operation plays the role of filtering, and that the resulting spectrum depends proportionally to the spectrum of the interpolating functions.

More specifically, the Fourier transform of a gaussian function $g(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$ is equal to $G(w_1, w_2) \triangleq \mathcal{FF}\{g(x, y)\} = e^{-\sigma^2(w_1^2+w_2^2)}$, which means that the spectrum of the gaussian interpolating function is still a gaussian function with bandwidth inversely proportional to σ^2 (variance). So if we choose a large σ , the interpolating function has a slowly decaying tail and behaves like a low-pass filter. Hence, we can get a smooth spectrum with low fluctuations. However, the high frequency components would be highly degraded. On the other hand, if we choose a small σ , the interpolating function approaches an impulse and behaves like a high-pass filter. However, in this case the evaluation of the Radon transform in some directions does not account for enough data points to fully reflect the nature of the desired spectrum, therefore, large fluctuations are likely to occur. Note that due to the bell shape of the spectrum of the gaussian interpolating function, low frequency components are expected to be less degraded and provide better resolution. If we have *a priori* information about the nature of the spectrum, we can choose a suitable σ accordingly.

In view of the effect of σ on the resulting spectrum, in the following we propose the following gaussian interpolating functions with different σ and normalization constants (c is a constant):

$$g(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}, \text{ with constant } \sigma \quad (15)$$

$$g(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}, \sigma = c \cdot i \text{ (} i \text{ denotes radius of the available data)} \quad (16)$$

$$g(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}, \sigma = c \cdot i' \text{ (} i' \text{ denotes radius of the data evaluated)} \quad (17)$$

$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}, \sigma = c \cdot i \text{ (} i \text{ denotes radius of the available data)} \quad (18)$$

The function (15) is the most basic one. The functions (16) and (17) take into account the fact that for data points farther away from the origin, the superposition effect using interpolation will not be

the same if we use a constant σ . If σ increases proportional to the radius, the interpolating function would decay slower as the radius increases, so that the weighting can be kept the same independent of the radius. The interpolating function of (18) is a normalized one in the sense that the weighting of the available data point is normalized to 1 as in the discretization case.

IV HIGH-RESOLUTION SPECTRAL ESTIMATION

We now focus on the gaussian interpolating functions, and use them to derive an analytically explicit procedure for spectral estimations with data points defined on a polar raster. Following the notations used in section III, we obtain

$$f(x, y) = \sum_{i=0}^N \sum_{m=1}^M f(i, m) e^{-[(x-r_i \cos \theta_m)^2 + (y-r_i \sin \theta_m)^2]/2\sigma^2} \quad (19)$$

where $r_i = i\delta_r$ and $\theta_m = 2\pi m/M$. Using the shifting property of the Radon transform and $\mathcal{R}\{e^{-(x^2+y^2)/2\sigma^2}\} = \sigma\sqrt{2\pi}e^{-t^2/2\sigma^2}$, it is straightforward to show that the exact Radon transform of (19) is

$$\hat{f}(t, \phi) = \mathcal{R}\{f(x, y)\} = \sigma\sqrt{2\pi} \sum_{i=0}^N \sum_{m=1}^M f(i, m) e^{-(t-r_i \cos(\theta_m - \phi))^2/2\sigma^2} \quad (20)$$

The complete procedure for estimating the power spectral density of a zero-mean homogeneous random field given discrete data $\{f(i, m), 0 \leq i \leq N, 1 \leq m \leq M\}$ and using the autocorrelation method of linear prediction is as follows:

1. Use (20) to compute the Radon transform of the data from $f(i, m)$, at some values of t with equal spacing and $\phi = 2\pi j/L, j = 1 \dots L$;
2. For each ϕ , compute the autocorrelation of the Radon transform; i.e., autocorrelate the results of (1) along each slice by (2) ($\mathbf{x} \triangleq (x, y)$)

$$r_\phi(\mathbf{x}) \triangleq \mathcal{R}_\phi\{r(\mathbf{x})\} = \mathcal{R}_\phi\{f(\mathbf{x}) ** f(-\mathbf{x})\} = R_\phi\{f(\mathbf{x})\} * R_\phi\{f(-\mathbf{x})\} = \hat{f}(t, \phi) * \hat{f}(t, -\phi) \quad (21)$$

3. For each ϕ , fit a 1-D AR(p) (p may vary for different ϕ) model to the projection data using the autocorrelation estimates, so we can get a set of linear prediction coefficients, say $\{h(k)\}$, corresponding to $r_\phi(j), j = 0, \dots, p-1$;

4. For each ϕ , use the 1-D AR(p) model to extend the covariance by [5]

$$r_{\phi}(j) = - \sum_{k=1}^{p-1} h(k)r_{\phi}(j-k), \quad j > p \quad (22)$$

5. Take 1-D Fourier transforms along each slice. This is the estimated 2-D spectral density.

Some comments are in order here:

1. The “autocorrelation” assumption of windowing data to zero for $i > N$ is required in order to compute the Radon transform of the data, since even $\hat{x}(0, \phi)$ depends on $\{x(t, \theta), t > N\}$;
2. It is therefore consistent to make the same assumption in fitting the 1-D AR models to each slice of the Radon transform;
3. As noted above, the Radon transform and autocorrelation operations commute, so the above procedure can properly be termed an “autocorrelation” procedure;
4. The covariance function of the interpolated function (19) is

$$\begin{aligned} r(u, v) &\triangleq \int_x \int_y f(x, y) f(x+u, y+v) dx dy \\ &= \sum_i \sum_j \sum_k \sum_l [f(r_i, \theta_j) f(r_k, \theta_l) \cdot e^{-[(u+r_i \cos \theta_j - r_k \cos \theta_l)^2 + (v+r_i \sin \theta_j - r_k \sin \theta_l)^2] / 4\sigma^2}] \end{aligned} \quad (23)$$

which is a Gaussian-weighted sum of the available discrete data sample – the weighting depends on the distance vector between two points. Equation (23) also provides a method to compute the covariance for data defined on a polar raster.

5. A significant advantage of this procedure is that it guarantees a psd covariance extension of the finite set of lag estimates computed from the data. This is required to ensure a non-negative power spectral density estimate;
6. Since the correlation matching property holds for the 1-D linear prediction technique along each slice of the Radon transform, it also must hold for the entire 2-D spectral estimate, in that

the 2-D covariance function derived from the 2-D spectral estimate will match the estimated covariance lags *in the Radon transform domain*.

V SIMULATION AND DISCUSSION

In this section we provide some examples to demonstrate the proposed spectral estimation procedure. The data are assumed to be available on a polar raster ($I \times M$, where I is the number of points along each direction (with radial spacing δ_r), and M is the number of angular partitions). A gaussian function is used as the interpolating function $\phi(x, y)$.

To compute the 2-D periodogram, we resample the interpolated data on a rectangular lattice (with spacing δ_x, δ_y along the abscissae and ordinate, respectively), zero-pad the points along each axis from L points to 128 points, and then take a 128×128 2-D discrete Fourier transform. To use the proposed new spectral estimation algorithm, we compute the Radon transform of the interpolated data, and then sample it on $I' \times M'$ polar raster, where I' is the number of points along each direction (with spacing $\delta_{r'}$), and M' is the number of angular partitions. The proposed spectral estimation procedure is then performed independently along each slice.

Note that in the following figures, the abscissae and ordinate denote the x and y axis for the 2-D periodograms, and radius and angles for the proposed method, respectively. For clarity, only one quadrant or one half of the spectrum is shown in the following figures. This is appropriate since the proposed method generates the spectral estimate slice-by-slice. However, the figures for the proposed method must be visually interpreted differently.

EXAMPLE 1

The algorithm of [17] was used to generate a single realization of a zero-mean isotropic random field with power spectrum density

$$S_1(w_1, w_2) = 4e^{-0.03(w_1^2 + w_2^2)}$$

which is a circularly symmetric spectrum as shown in Figure 2. The available data was 3×6 ($I =$

3, $M = 6$) with radial spacing $\delta_r = 0.2$. We used $\delta_x = \delta_y = \delta_{r'} = 0.1$, $L = 25$, $I' = 6$, $M' = 36$, and chose the interpolating function defined in (16) with $\sigma = 0.15i$.

The resulting spectral estimates are shown in in Figures 3 (for the 2-D periodogram) and 4 (for the proposed method with AR(4) modeling along each slice). The estimated spectra in both figures are similar, and close to the true spectrum.

EXAMPLE 2a

The algorithm of [17] was used to generate a single realization of an isotropic random field with power spectrum density

$$S_2(w_1, w_2) = \begin{cases} 10 & \text{if } w_1^2 + w_2^2 \leq (0.645\pi)^2 \\ 0 & \text{otherwise} \end{cases}$$

which is plotted in Figure 5. The available data had $I = 3$ and $M = 6$ with radial spacing $\delta_r = 0.2$. We used $\delta_x = \delta_y = \delta_{r'} = 0.1$, $L = 31$, $I' = 12$, $M' = 72$, and the interpolating function defined in (16).

The resulting spectral estimates are shown in Figures 6 (2-D periodogram) and 7 (spectrum using the proposed method). Note that the proposed procedure provides better transition performance on the discontinuity of the original spectrum. However, the 1-D extrapolation of the 1-D covariance also causes a slight increase of the high frequency components in Figure 7.

EXAMPLE 2b

The algorithm of [17] was used to generate a single realization of an isotropic random field with power spectrum density $S_2(w_1, w_2)$ (same as for Example 2a) plus a white gaussian noise field at a SNR equal to 7dB. Now $\sigma = 0.3i$ is used in the interpolating function (16); all other parameters are the same as in Example 2a.

The resulting spectral estimates of power spectral density are shown in Figures 8 (2-D periodogram) and 9 (spectrum using the proposed method), respectively. Note that the estimated spectrum in Figure 9 is not as good as that in Figure 7, due to the additive white noise. However, it is still better than the 2-D periodogram shown in Figure 8.

EXAMPLE 3a

Here the random field whose power spectral density is to be estimated is the *deterministic* 2-D signal

$$D_1(x, y) = \cos(w_1x + w_2y) + \cos(w_3x + w_4y)$$

where $(w_1, w_2) = (0.173\pi, 0.1\pi)$, $(w_3, w_4) = (0.12\pi, 0.208\pi)$, $x = i\delta_r \cos(j\theta)$, $y = i\delta_r \sin(j\theta)$, $0 \leq i \leq I$, $1 \leq j \leq M$, and $\theta = 2\pi/12$ ($M=12$). This consists two closely-spaced low frequency sinusoids. The available data has $I = 12$ and $M = 12$ with radial spacing $\delta_r = 1$. We used $\delta_x = \delta_y = \delta_{r'} = 1$, $L = 31$, $I' = 12$, $M' = 72$, and the normalized interpolating function defined in (18). σ is chosen to be $0.02i$, which is much smaller than the spacings of the interpolated points, so the interpolating function is close to an impulse function. This is a reasonable choice; since if σ is too large, the spectrum will be smeared by those of the adjacent directions, which will reduce the overall resolution. An AR(3) model is used to extrapolate the 1-D covariances in the proposed method.

The resulting spectral estimates are shown in Figures 10 (2-D periodogram) and 11 (spectrum using the proposed method). Note that the 2-D periodogram in Figure 10 shows only one peak, so that it fails to resolve two sinusoids. In contrast, for the proposed method in Figure 11, two peaks are apparent and are located at $(0.175\pi, 0.109\pi)$ and $(0.117\pi, 0.203\pi)$, respectively, which are very close to the true frequencies. More accurate results were achieved using more points along each direction. Also note that the artifacts in Figure 10 are exaggerated in appearance, due to the nature of the plotting axes. A radial, rather than rectangular, plot of axes radius r vs. angle θ would reduce the visibility of the artifacts.

EXAMPLE 3b

Here the random field whose power spectral density is to be estimated consists of the deterministic signal from Example 3a plus a single realization of a white gaussian noise field with unit power:

$$D_2(x, y) = \cos(w_1x + w_2y) + \cos(w_3x + w_4y) + w(x, y)$$

We use the normalized interpolating function defined in (18) with $\sigma = 0.02i$, and AR(3) modeling

along each slice in the Radon transform domain. All other parameters are the same as in Example 3a.

The resulting spectral estimates are shown in Figures 12 (2-D periodogram) and 13 (proposed method). Although some spurious peaks appear in Figure 13, due to the additive white noise, the two peaks for the sinusoidal input signals are still obviously distinguishable in Figure 13. The 2-D periodogram in Figure 12 not only contains many spurious peaks, but also fails to resolve two sinusoids.

Use of a Bessel function as the interpolating function gave less satisfactory results; in the resulting spectral estimate the two sinusoidal peaks are not resolved.

VI 2-D CORRELATION MATCHING ON A POLAR RASTER

A. Introduction

The above 2-D spectral estimation method may be used to estimate 2-D spectra on a polar raster, either directly from data or from specified covariance lags. In the latter case, however, the above method does not preserve the specified covariance lags: The inverse 2-D Fourier transform of the 2-D power spectral density (the 2-D covariance) does not match the given covariance lags. Hence it does not satisfy correlation matching in the 2-D plane.

In this section we propose a procedure that *extends* a given set of 2-D covariance lags, specified inside a disk of radius R , into a 2-D covariance function specified everywhere in \mathcal{R}^2 , and which matches the given 2-D covariance lags. It guarantees that the extended covariance is a 2-D psd (positive semi-definite) function, ensuring that the power spectral density will be non-negative everywhere. Although the procedure is applied to functions defined continuously on \mathcal{R}^2 , it may also be applied to discrete covariance lags on a polar raster by interpolation, as described above. We also discuss when such an extension is impossible, and how this is manifested in the algorithm.

The problem addressed is as follows:

Covariance Extension Problem: Given a set of covariance lags $\{f(r, \theta), r < R\}$ for some radius R , determine an *extension* $\{f(r, \theta), r > R\}$ of the given lags such that: (1) $f(r, \theta)$ agrees with the given

values $\{f(r, \theta), r < R\}$; and (2) $f(r, \theta)$ is a 2-D psd function, meaning that its 2-D Fourier transform is real and non-negative everywhere.

B. Radon and Backprojection Transforms

To explain the procedure, and to explain why it is necessary for covariance extension, we define the Radon transform, the backprojection transform, and note some causality and psd-preserving properties of each transform.

Radon Transform in Polar Coordinate: Let $f(x) = f(r, \theta)$ be a function defined on $x \in \mathcal{R}^2$. Then the Radon transform $\hat{f}(t, \phi)$ of $f(r, \theta)$ is

$$\hat{f}(t, \phi) = \mathcal{R}\{f(r, \theta)\} = \int_0^\infty \int_0^{2\pi} f(r, \theta) \delta(t - r \cos(\theta - \phi)) r d\theta dr \quad (24)$$

Note that the Radon transform is the *line integral* of $f(r, \theta)$ along the line $t = x \cos \phi + y \sin \phi$, where $x = r \cos \theta$ and $y = r \sin \theta$.

Backprojection Transform: Let $\hat{f}(x) = f(r, \theta)$ be a function defined on $x \in \mathcal{R}^2$. Then the backprojection transform $\hat{f}(t, \phi)$ of $f(r, \theta)$ is

$$\begin{aligned} \hat{f}(t, \phi) = \mathcal{B}\{f(r, \theta)\} &= \int_0^\infty \int_0^\pi f(r, \theta) \delta(r - t \cos(\theta - \phi)) d\theta dr \\ &= \int_0^\pi f(r = t \cos(\theta - \phi), \theta) d\theta \end{aligned} \quad (25)$$

Note that the backprojection transform is the *circular mean* of $f(r, \theta)$ on the circle $r = t \cos(\theta - \phi)$ (note that the point (r, θ) coincides with the point $(-r, \theta + \pi)$; this is why the integral over θ varies only from 0 to π rather than 2π). This circle passes through the origin, has diameter t , and has its center at $((t/2)\cos \phi, (t/2)\sin \phi)$. The backprojection transform is also half the *adjoint* of the Radon transform ([12], p.134); note that (24) and (25) differ primarily in that r and t have been interchanged.

Anticausality of Radon Transform: Let $\hat{f}(t, \phi) = \mathcal{R}\{f(r, \theta)\}$. Then for any $T > 0$, $\hat{f}(T, \phi)$ depends *only* on the values $\{f(r, \theta), r > T\}$. This is clear since $\hat{f}(T, \theta)$ is the line integral of $f(\cdot)$ along the line $T = x \cos \theta + y \sin \theta$, whose minimum distance from the origin is T . It is also true that given

$\{\hat{f}(t, \phi), t > T\}$, it is possible to reconstruct $\{f(r, \theta), r > T\}$; an explicit formula has been given by Cormack [18].

This anticausality explains why the above spectral estimation procedure does not preserve the given covariance lags. Any given covariance lag at radius T depends on all values $t > T$ of the Radon transform of the covariance. But these values for $t > R$ have been changed from zero by the 1-D extensions applied to $\hat{f}(t, \theta)$ independently for each θ . Hence the extended covariance does not match the given covariance lag.

Causality of Backprojection Transform: Let $\hat{f}(t, \phi) = \mathcal{B}\{f(r, \theta)\}$. Then for any $T > 0$, $\hat{f}(T, \phi)$ depends *only* on the values $\{f(r, \theta), r \leq T\}$. This is clear since $\hat{f}(T, \theta)$ is the circular mean of $f(\cdot)$ along the circle $r = T \cos(\theta - \phi)$, so that $r \leq T$ always. Another way to see this is to note that backprojection at the point (T, ϕ) can also be viewed as the integration over all lines $r = x \cos \theta + y \sin \theta$ passing through (T, ϕ) ; any such line must pass closer to the origin than T , so that any such line will have $r < T$. It is also true that given $\{\hat{f}(t, \phi), t < T\}$, it is possible to reconstruct $\{f(r, \theta), r < T\}$ (see [19]).

Inverse Radon Transform by Backprojection: Let $\hat{f}(t, \phi) = \mathcal{R}\{f(r, \theta)\}$. Then we may recover $f(r, \theta)$ from $\hat{f}(t, \phi)$ by computing

$$f(r, \theta) = \mathcal{B}_{(t, \phi) \rightarrow (r, \theta)} \mathcal{H} \frac{d}{dt} \hat{f}(t, \phi) \quad (26)$$

where \mathcal{H} denotes the Hilbert transform $\mathcal{H}\{f(t)\} = f(t) * \frac{1}{\pi t}$. This is the well-known technique of *filtered backprojection* [12]. Note that here $\hat{f}(t, \phi)$ is regarded as a collection of functions indexed by ϕ , rather than as a continuous function of polar coordinates (t, ϕ) .

Positive Semi-Definite Properties of \mathcal{R} and \mathcal{B} : Let $f(r, \theta)$ be a 2-D psd function. Then $\hat{f}(t, \phi) = \mathcal{R}\{f(r, \theta)\}$ is a 1-D psd function in t for each ϕ by the projection-slice theorem of the Radon transform, and $\mathcal{H} \frac{d}{dt} \hat{f}(t, \phi)$ is also a 1-D psd function in t for each ϕ , since the filtering operation $\mathcal{H} \frac{d}{dt}$ corresponds to multiplication by $|k|$ in the Fourier domain: $\mathcal{F}_{t \rightarrow k} \{\mathcal{H} \frac{d}{dt} \hat{f}(t, \phi)\} = |k| \mathcal{F}_{t \rightarrow k} \{\hat{f}(t, \phi)\}$. Hence the inverse backprojection transform \mathcal{B}^{-1} maps 1-D psd functions to 2-D psd functions, as does the inverse Radon

transform.

C. Covariance Extension Procedure

We propose the following procedure for extending a given set of covariance lags $f(x), x \in \mathcal{R}^2, |x| < R$ into a function $f(x), x \in \mathcal{R}^2$ specified everywhere in \mathcal{R}^2 and which agrees with the given set of covariance lags:

1. Compute the Radon transform $\hat{f}(t, \phi)$ of the function $f(r, \theta)$ defined by $f(r, \theta) = f(x)$ if $r = |x| < R$; 0 if $r > R$. Note that $\hat{f}(t, \phi) = 0$ for $t > R$ by anticausality of the Radon transform;
2. Compute $\mathcal{H} \frac{d}{dt} \hat{f}(t, \phi)$ from $\hat{f}(t, \phi)$. Note that $\mathcal{H} \frac{d}{dt} \hat{f}(t, \phi) \neq 0$ for $t > R$ due to the smearing effect of the Hilbert transform \mathcal{H} ;
3. Replace the values of $\mathcal{H} \frac{d}{dt} \hat{f}(t, \phi)$ for $t > R$ with values such that $\mathcal{H} \frac{d}{dt} \hat{f}(t, \phi)$ is 1-D psd in t for each ϕ . Call this new function $\mathcal{H} \frac{d}{dt} \tilde{f}(t, \phi)$; note that $\mathcal{H} \frac{d}{dt} \tilde{f}(t, \phi) = \mathcal{H} \frac{d}{dt} \hat{f}(t, \phi)$ for $t < R$;
4. Compute $F(r, \theta) = \mathcal{B}_{(t, \phi) \rightarrow (r, \theta)}^{-1} \{ \mathcal{H} \frac{d}{dt} \tilde{f}(t, \phi) \}$. $F(r, \theta)$ is the 2-D psd extended covariance function.

By the causality property of \mathcal{B} , $F(r, \theta) = f(r, \theta)$ for $r < R$, so that the extended covariance matches the given covariance lags. By the psd property of \mathcal{B} , $F(r, \theta)$ is a 2-D psd function since $\mathcal{H} \frac{d}{dt} \tilde{f}(t, \phi)$ is a 1-D psd function in t for each ϕ . Hence we have successfully extended the given covariance lags into a 2-D psd covariance function $F(r, \theta)$.

Note that the only difference between this procedure and the previous procedure is that $\mathcal{H} \frac{d}{dt}$ of the Radon transform of the given covariance lags is computed *before* performing the 1-D psd extensions, instead of *after*. This seemingly minor change allows the use of the causality property of \mathcal{B} , instead of the anticausality property of \mathcal{R}^{-1} .

It might seem at first glance that this constructive procedure allows *any* 2-D set of covariance lags specified inside a disk of radius R to be extended into a 2-D psd covariance function. This seems to contradict the known fact [7] that some sets of covariance lags are *not* extendible into a 2-D psd covariance function. The resolution of this paradox is found by noting that it may not be possible to

form a 1-D psd function $\mathcal{H}\frac{d}{dt}\hat{f}(t, \phi)$ from the $\mathcal{H}\frac{d}{dt}\hat{f}(t, \phi)$ computed from the given covariance lags. For example, if for any ϕ there is a $t < R$ such that $\mathcal{H}\frac{d}{dt}\hat{f}(t, \phi) > \mathcal{H}\frac{d}{dt}\hat{f}(0, \phi)$, then the 1-D psd extension cannot be performed for that ϕ , since *any* psd function $g(t)$ must have the property that $g(t) \leq g(0)$. This explains how a 2-D extension may be impossible.

Another important question is: Can *all* possible 2-D psd extensions of the given $f(x), x \in \mathcal{R}^2, |x| < R$ be found from all of the possible 1-D psd extensions of the $\mathcal{H}\frac{d}{dt}\hat{f}(t, \phi)$? Unfortunately, the answer is no. To see why, we now investigate briefly the nullspace of \mathcal{B} .

D. Nullspace of Backprojection Operator

Let $f(x), x \in \mathcal{R}^2, |x| > R$ be an extension of given values $f(x), |x| < R$. Now compute the filtered Radon transforms of both the given values $f(x), |x| < R$ and the extended values $f(x), |x| > R$ (note the latter is a “hollow” function):

$$\begin{aligned}\hat{f}_{ext}(t, \phi) &= \mathcal{H}\frac{d}{dt}\mathcal{R}\{f(x), |x| > R; 0, |x| < R\}; \\ \hat{f}_{int}(t, \phi) &= \mathcal{H}\frac{d}{dt}\mathcal{R}\{f(x), |x| < R; 0, |x| > R\}\end{aligned}$$

Here $\hat{f}_{int}(t, \phi)$ is the function which is extended to create a 1-D psd in the procedure we proposed above, and by construction, $\mathcal{B}\{\hat{f}_{ext}(t, \phi)\} = 0$ for $t < R$.

We now consider the following question: Does $\hat{f}_{ext}(t, \phi) = 0$ for $t < R$? That is, is there a non-zero function $\hat{f}_{ext}(t, \phi)$ such that $\mathcal{B}\{\hat{f}_{ext}(t, \phi)\} = 0$, i.e., does \mathcal{B} have a non-empty nullspace?

The significance of this question is as follows. If \mathcal{B} does NOT have a non-empty nullspace, then $\hat{f}_{ext}(t, \phi) = 0$ for $t < R$. Then

$$\mathcal{H}\frac{d}{dt}\mathcal{R}\{f(x)\} = \hat{f}_{ext}(t, \phi) + \hat{f}_{int}(t, \phi) = \hat{f}_{int}(t, \phi), t < R$$

and ANY extension of given values $f(x), |x| < R$ is associated with an extension of $\hat{f}_{int}(t, \phi)$, so that ALL 2-D psd extensions of the given lags are associated with 1-D psd extensions of $\hat{f}_{int}(t, \phi)$. But if \mathcal{B} HAS a non-empty nullspace containing some non-zero $\hat{f}_{ext}(t, \phi)$, then

$$\mathcal{H}\frac{d}{dt}\mathcal{R}\{f(x)\} = \hat{f}_{ext}(t, \phi) + \hat{f}_{int}(t, \phi) \neq \hat{f}_{int}(t, \phi), t < R$$

so that the extended $f(x)$ is NOT associated with 1-D extensions of $\hat{f}_{int}(t, \phi), t < R$, but with 1-D extensions of $\hat{f}_{int}(t, \phi) + \hat{f}_{int}(t, \phi), t < R$. This implies that not all 2-D extensions of $f(x), |x| < R$ can be found from 1-D extensions of $\hat{f}_{int}(t, \phi), t < R$.

Unfortunately, \mathcal{B} DOES have a non-empty nullspace, so that it is not true that all 2-D psd extensions of a given set of covariance lags can be found using the procedure proposed above. Indeed, it might seem that for ANY function $f(x)$ such that $f(x) = 0, |x| < R$, we would have $\mathcal{H} \frac{d}{dt} \mathcal{R}\{f(x)\} \neq 0$ for $t < R$. Of course this is not true—indeed, our procedure constructs functions $f(x) = 0, |x| < R$ such that $\mathcal{H} \frac{d}{dt} \mathcal{R}\{f(x)\} = 0$ for $t < R$. Furthermore, we have the following theorem:

Theorem:

Let $f(r, \theta) = 0$ for $r < R$ and let $\mathcal{H} \frac{d}{dt} \mathcal{R}\{f(r, \theta)\} = 0$ for $t < \epsilon$, for any $\epsilon > 0$. Then $\mathcal{H} \frac{d}{dt} \mathcal{R}\{f(r, \theta)\} = 0$ for all $t < R$.

Proof: To prove this theorem we need the following lemma:

Lemma:

Let $g(r, \theta)$ be any continuous function equaling zero at the origin. Define $g'(r, \theta) = g(R/r, \theta)$. compute the Radon transform $\hat{g}'(t, \phi)$ of $g'(r, \theta)$, and define $\hat{g}(t, \phi) = \hat{g}'(R/t, \phi)$. Then $\hat{g}(t, \phi) = t \mathcal{B}_{(r, \theta) \rightarrow (t, \phi)} \{g(r, \theta)/r^2\}$.

Proof of Lemma: We have

$$\begin{aligned} \hat{g}(t, \phi) &= \mathcal{R}\{g(R/r, \theta)\}_{t \rightarrow R/t} = \int_0^\infty \int_0^{2\pi} \delta(R/t - r \cos(\theta - \phi)) g(R/r, \theta) r d\theta dr \\ &= \int_0^\infty \int_0^{2\pi} \delta(R/t - R/r \cos(\theta - \phi)) g(r, \theta) R^2 r^{-3} d\theta dr \\ &= \int_0^\infty \int_0^{2\pi} \delta(r - t \cos(\theta - \phi)) g(r, \theta) t R r^{-2} d\theta dr \\ &= t R \mathcal{B}_{(r, \theta) \rightarrow (t, \phi)} \{g(r, \theta)/r^2\} \end{aligned} \quad (27)$$

where we have changed variables from r to R/r and used the scaling property $\delta(xR/(rt)) = rt/R\delta(x)$ of the impulse. ■

This result is not surprising: Reflecting a function across the circle of radius R amounts to taking its *involute*, and the involute of a line (along which the Radon transform is computed) is a circle (along

which the circular mean, i.e., the backprojection, is computed).

Proof of Theorem: For convenience in using the Lemma switch the variables t and r , and ϕ and θ .

Define $g(t, \phi) = Rtf(t, \phi)$. Then

$$\mathcal{H} \frac{d}{dr} \mathcal{R}\{f(t, \phi)\} = \mathcal{B}^{-1}\{g(t, \phi)/(Rt)\} = \check{g}(r, \theta)/r^2$$

where $\check{g}(r, \theta) = \mathcal{R}^{-1}\{g(R/t, \phi)\}_{r \rightarrow R/t}$. But we are given that $g(t, \phi) = Rtf(t, \phi) = 0$ for $t < R$, which implies that $g(R/t, \phi) = 0$ for $t > 1$. But then $\mathcal{R}^{-1}\{g(R/t, \phi)\} = 0$ for $r > 1$, so that $\check{g}(r, \theta) = \mathcal{R}^{-1}\{g(R/t, \phi)\}_{r \rightarrow R/t} = 0$ for $r < R$. The result follows immediately. ■

The heart of the above proof is the conclusion that $g(R/t, \phi) = 0$ for $t > 1$ implies that $\mathcal{R}^{-1}\{g(R/t, \phi)\} = 0$ for $r > 1$. Although this seems obvious, it is not in fact true unless $\mathcal{R}^{-1}\{g(R/t, \phi)\}$ is *also* known to go to zero sufficiently fast as $r \rightarrow \infty$. This is why we also need the condition $\check{g}(r, \theta) = 0$ for $r < \epsilon$, so that $\check{g}(r, \theta) = \mathcal{R}^{-1}\{g(R/t, \phi)\}_{r \rightarrow R/t}$ is known to be zero for $r > R/\epsilon$.

The major point of this section is that the inability of our proposed procedure to specify all of the 2-D psd extensions of the given covariance lags, due to the non-empty nullspace of \mathcal{B} , is not as bad as it may first appear.

VII CONCLUSION

A procedure for estimating the power spectral density of a homogeneous random field from discrete data inside a disk of finite radius has been presented. Unlike spectral density estimators using 2-D linear prediction on a rectangular raster, the estimated spectral density is guaranteed to be non-negative, since the extended (in the Radon transform domain) covariance is guaranteed to be psd.

The procedure operates by employing a novel interpolation technique, using gaussian basis functions to compute the Radon transform analytically from a few discrete data points. 1-D linear prediction is then used along each slice to compute spectral density estimates along each slice of the Radon transform. The procedure can be viewed as an “autocorrelation” method, since the unknown data is windowed to zero both for purposes of computing the Radon transform and for fitting the 1-D AR

models to each slice of the Radon transform. This procedure also provide a high-resolution spectral estimates for the data on the polar raster. Some numerical examples are provided to demonstrate the validity of this procedure.

ACKNOWLEDGMENT

The work of both authors was supported by the Air Force Office of Scientific Research under grant # AFOSR-89-0017.

References

- [1] G.T. Herman ed., *Image Reconstruction from Projections: Implementation and Applications*, Springer-Verlag, NY, 1979.
- [2] J.L. Walker, *Range-Doppler Imaging of Rotating Objects*, IEEE Trans. Aerosp. Electron. Syst. vol. AES-16, pp. 23-52, Jan. 1980.
- [3] A.K. Jain and S. Ranganath, *Extrapolation Algorithms for Discrete Signals with Application in Spectral Estimation*, IEEE Trans. ASSP, Vol. ASSP-29, No. 4, pp. 830-845, April 1981.
- [4] L.S. Joyce, *A Separable 2-D Autoregressive Spectral Estimation Algorithm*, Proc. 1979 ICASSP, Washington DC, April 1979, pp. 673-676.
- [5] S.M. Kay, *Modern Spectral Estimation*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [6] N. Srinivasa, K.R. Ramakrishnan, and K. Rajgopal, *Two-Dimensional Spectral Estimation: A Radon Transform Approach*, IEEE J. Ocean Eng. Vol. OE-12, No. 1, pp. 90-96, Jan. 1987.
- [7] T. Marzetta, *A Linear Prediction Approach to Two-Dimensional Spectral Factorization and Spectral Estimation*, Ph.D. Thesis, Dept. of EECS, M.I.T., Cambridge, MA, Feb. 1978.
- [8] P. Maragos, R.W. Schafer, and R.M. Mersereau, *Two-Dimensional Linear Prediction and its Application to Adaptive Coding of Images*, IEEE Trans. ASSP, Vol. ASSP-32, No.6, pp. 1213-1229, June 1984.

- [9] B. Dickinson, *Two-Dimensional Markov Spectrum Estimates Need Not Exist*, IEEE Trans. Info. Th. Vol. IT-26, No. 1, pp. 120-121, Jan. 1980.
- [10] A.H. Tewfik, B.C. Levy, and A.S. Willsky, *Sampling Theorems for Two-Dimensional Isotropic Random Fields*, IEEE Trans. Info. Th. Vol. IT-34, No. 5, pp. 1092-1096, Sept. 1988.
- [11] H. Stark, *Sampling Theorems in Polar Coordinates*, J. Opt. Soc. Am., Vol. 69, pp.1519-1525, Nov. 1979.
- [12] S.R. Deans, *The Radon Transform and Some of its Applications*, John Wiley and Sons Inc, New York, 1983.
- [13] I. Gertner, *A New Efficient Algorithm to Compute the Two-Dimensional Discrete Fourier Transform*, IEEE Trans. ASSP, Vol. ASSP-36, No. 7, pp. 1036-1050, July 1988.
- [14] A.K. Jain and S. Ansari, *Radon Transform Theory for Random Fields and Optimum Image Reconstruction from Noisy Data*, Proc. 1984 ICASSP, March 1984.
- [15] N. Ohyama *et al*, *Discrete Radon Transform in a Continuous Space*, J. Opt. Soc. Am., Vol. 4, pp. 318-324, Feb. 1987.
- [16] G. Beylkin, *Discrete Radon Transform*, IEEE Trans. ASSP, Vol. ASSP-35, No. 2, pp. 162-172, Feb. 1987.
- [17] M. Shinozuka and C.-M. Jan, *Digital Simulation of Random Processes and its Applications*, J. Sound and Vibration, Vol. 25, No.1, pp. 111-128, Jan. 1972.
- [18] A. M. Cormack, *Representation of a Function by its Line Integrals, with Some Radiological Applications II*, J. Appl. Phys., Vol. 35, pp. 2908-2913, 1964.
- [19] H. Hellsten and L. E. Andersson, *An Inverse Method for the Processing of Synthetic Aperture Radar Data*, Inverse Problems, Vol. 3, pp. 111-124, 1987.

FIGURE HEADING

1. Figure 1: The polar raster on which the 2-D random field is defined with $M = 8$.
2. Figure 2: Spectrum of $S_1(w_1, w_2) = 4e^{-0.03(w_1^2 + w_2^2)}$.
3. Figure 3: 2-D periodogram for Example 1.
4. Figure 4: Spectrum obtained by using the covariance extension for Example 1.
5. Figure 5: Spectrum of $S_2(w_1, w_2) = 10$ if $w_1^2 + w_2^2 \leq (0.645\pi)^2$, $= 0$ otherwise.
6. Figure 6: 2-D periodogram for Example 2a.
7. Figure 7: Spectrum using the proposed method for Example 2a.
8. Figure 8: 2-D periodogram for Example 2b.
9. Figure 9: Spectrum using the proposed method for Example 2b.
10. Figure 10: 2-D periodogram for Example 3a using normalized interpolating function.
11. Figure 11: Spectrum for Example 3a using normalized interpolating function and the proposed method.
12. Figure 12: 2-D periodogram for Example 3b using normalized interpolating function.
13. Figure 13: Spectrum for Example 3b using normalized interpolating function and the proposed method.

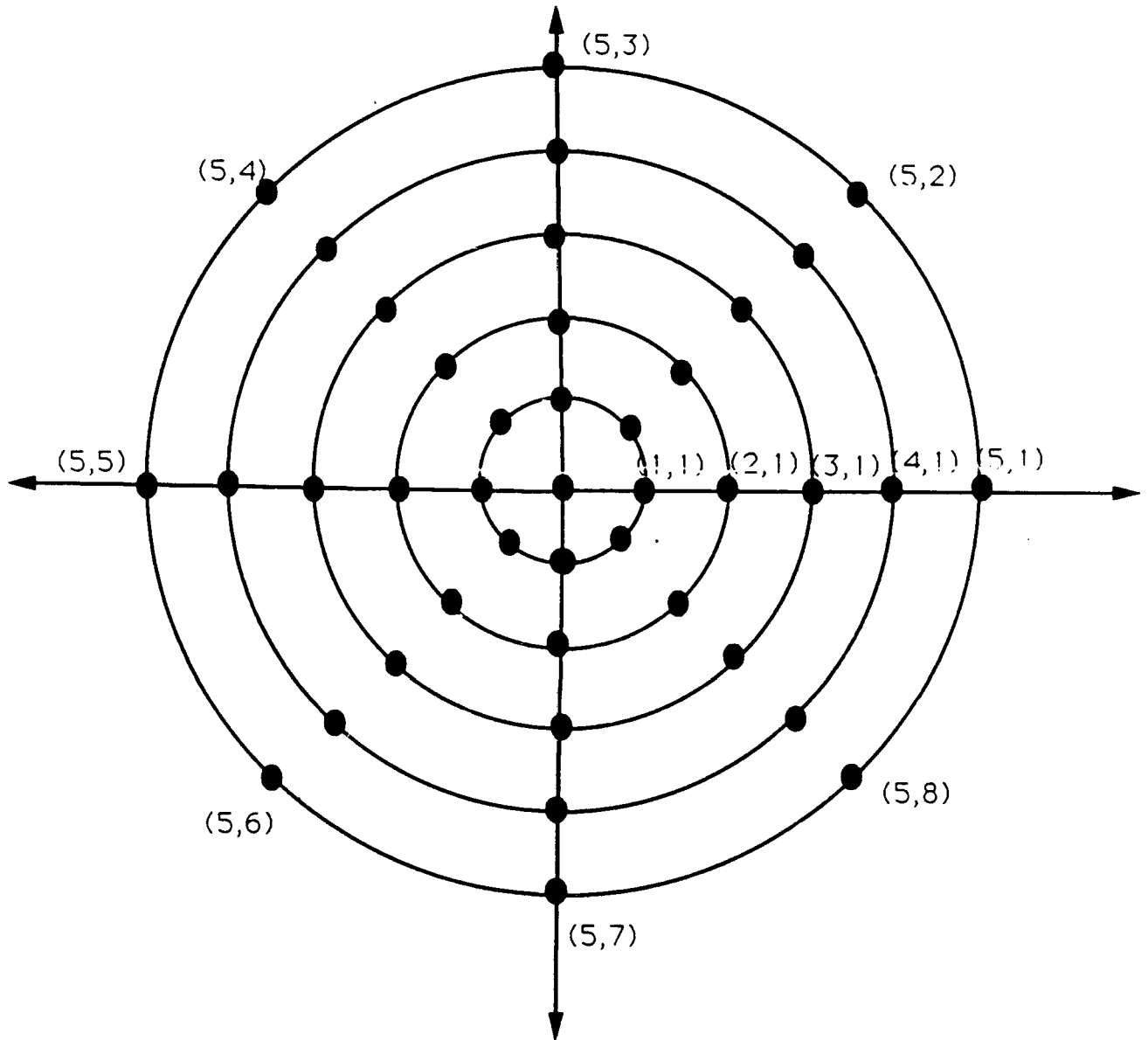


Figure 1: The polar raster on which the 2-D random field is defined with $M = 8$.

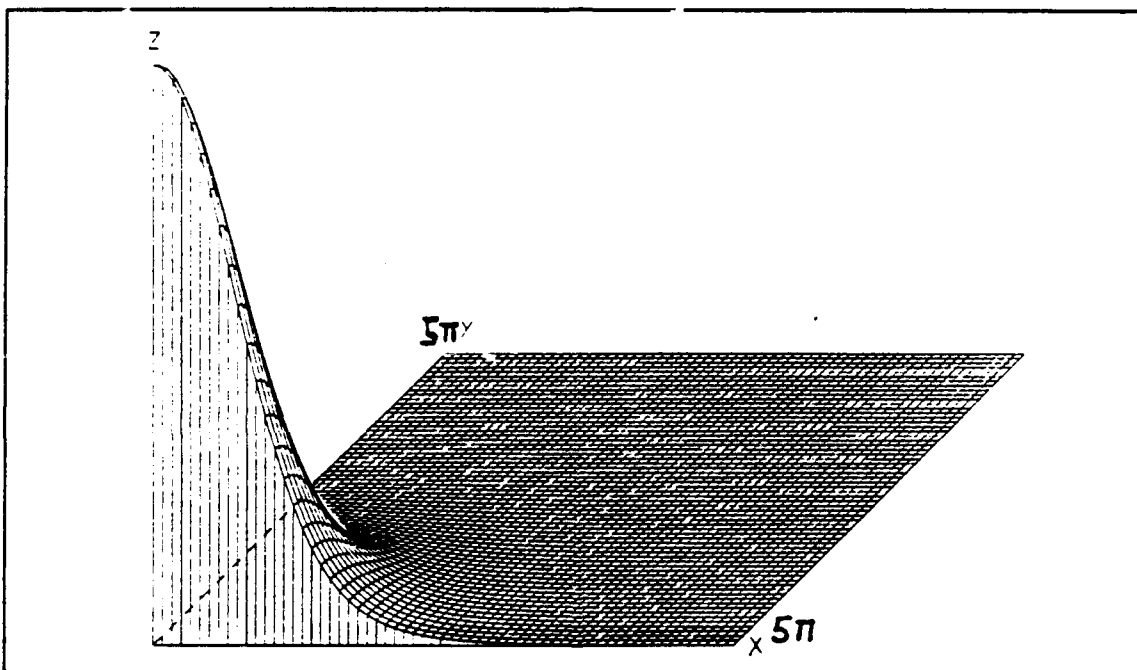


Figure 2: Spectrum of $S_1(w_1, w_2) = 4e^{-0.03(w_1^2 + w_2^2)}$.

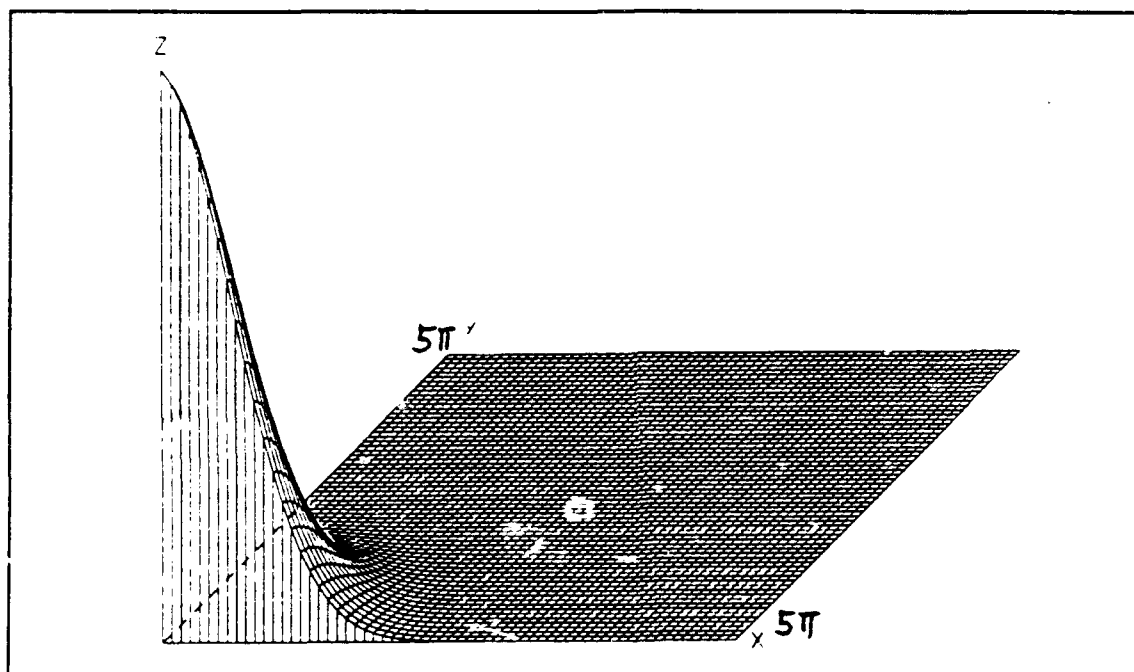


Figure 3: 2-D periodogram for Example 1.

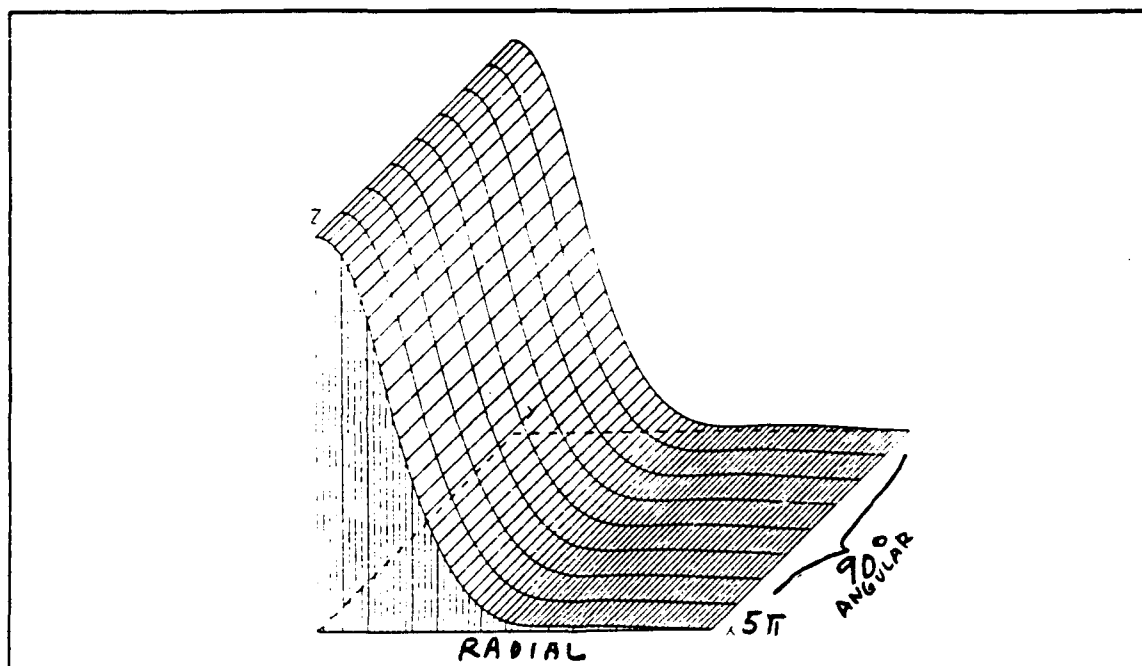


Figure 4: Spectrum obtained by using the covariance extension for Example 1.

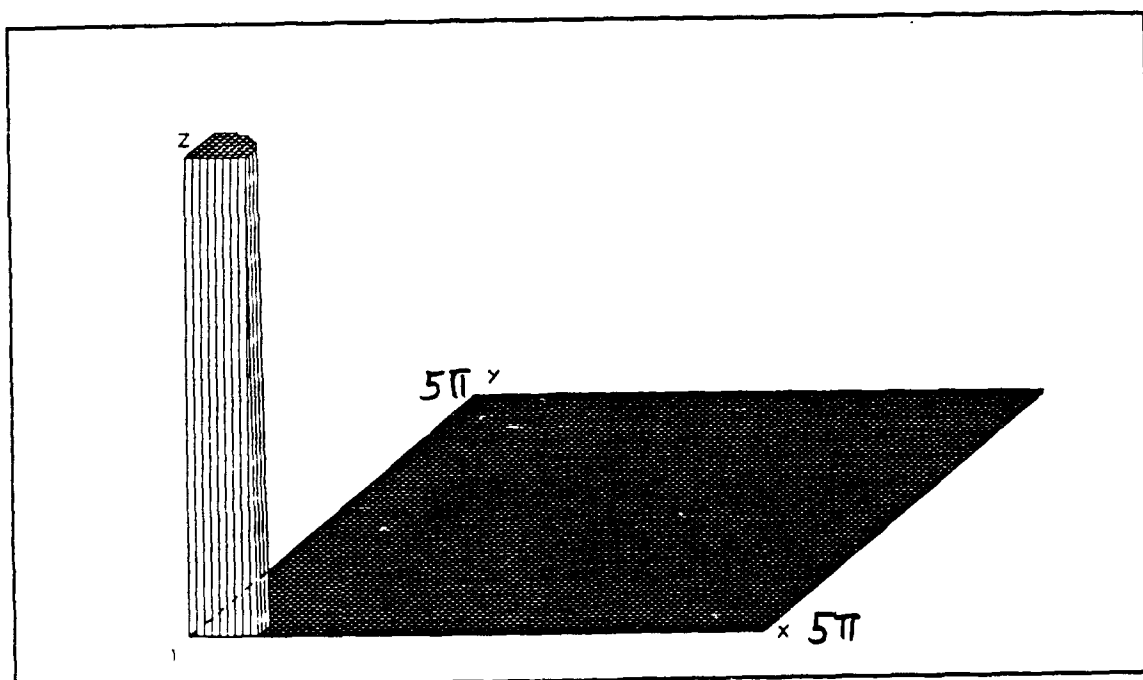


Figure 5: Spectrum of $S_2(w_1, w_2) = 10$ if $w_1^2 + w_2^2 \leq (0.645\pi)^2$, $= 0$ otherwise.

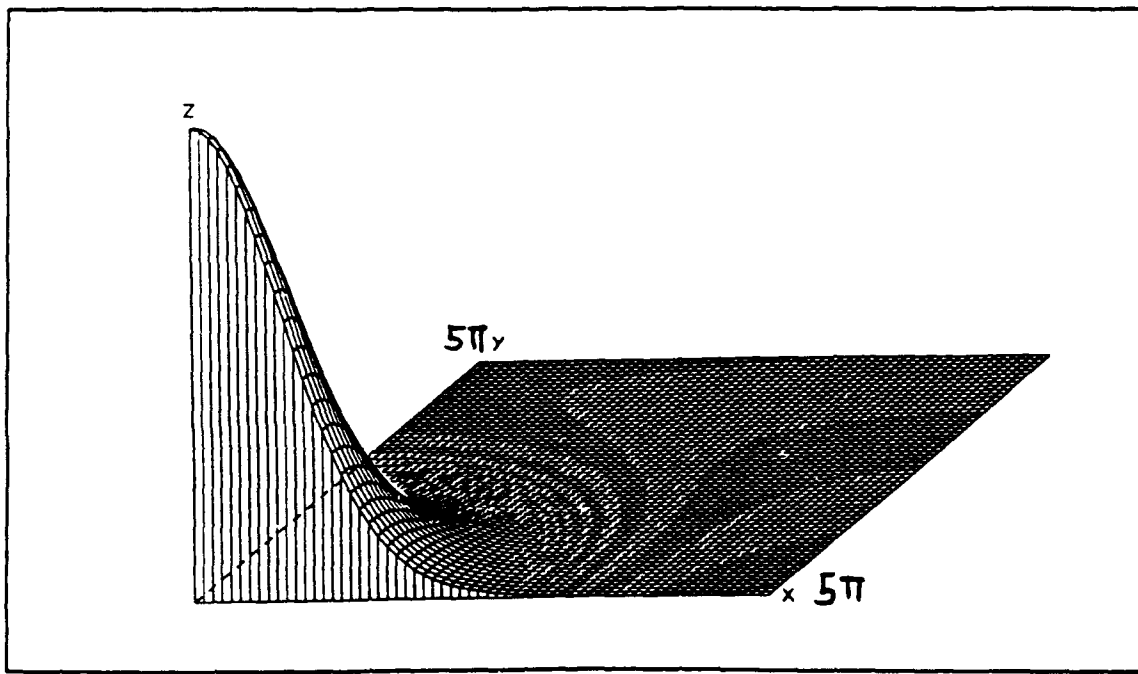


Figure 6: 2-D periodogram for Example 2a.

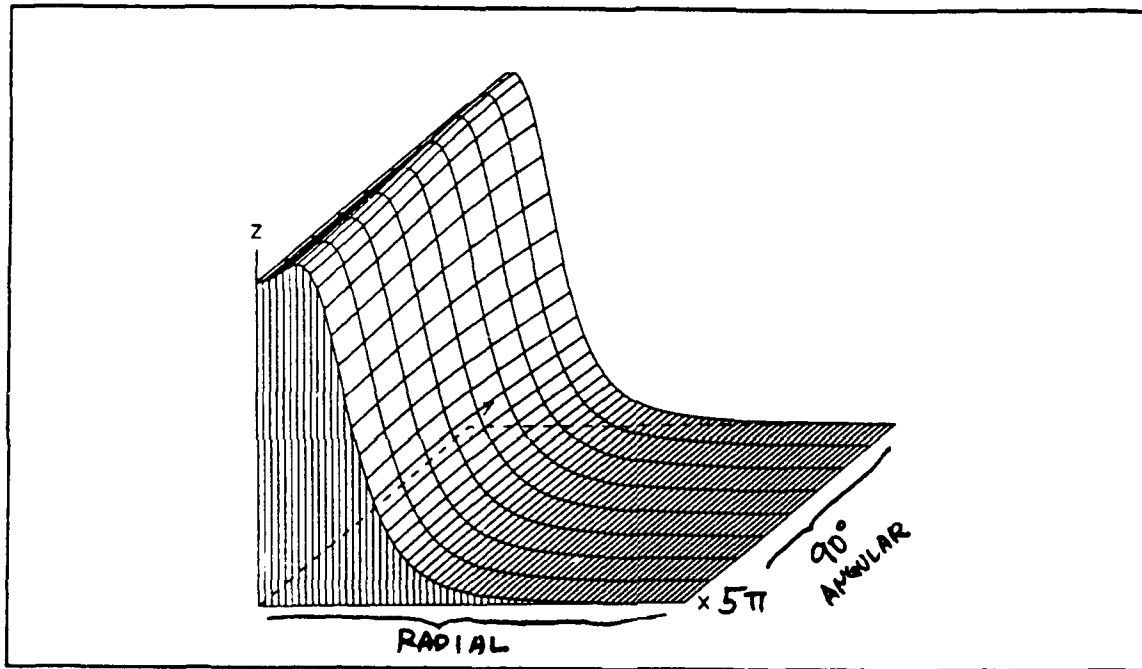


Figure 7: Spectrum using the proposed method for Example 2a.

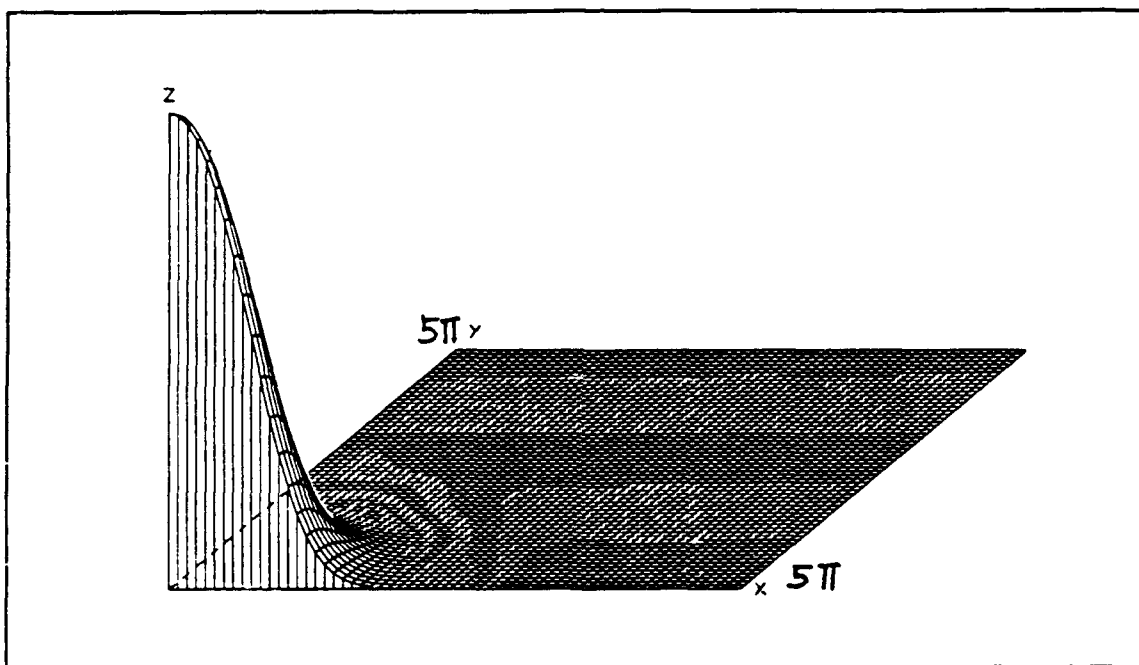


Figure 8: 2-D periodogram for Example 2b.

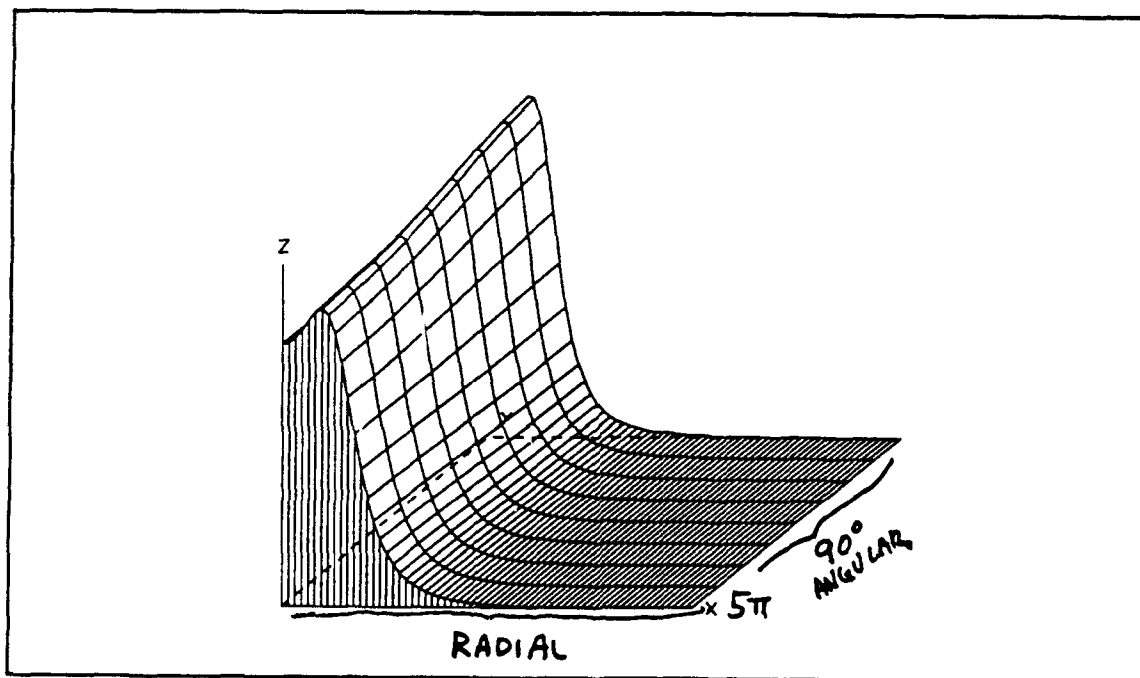


Figure 9: Spectrum using the proposed method for Example 2b.

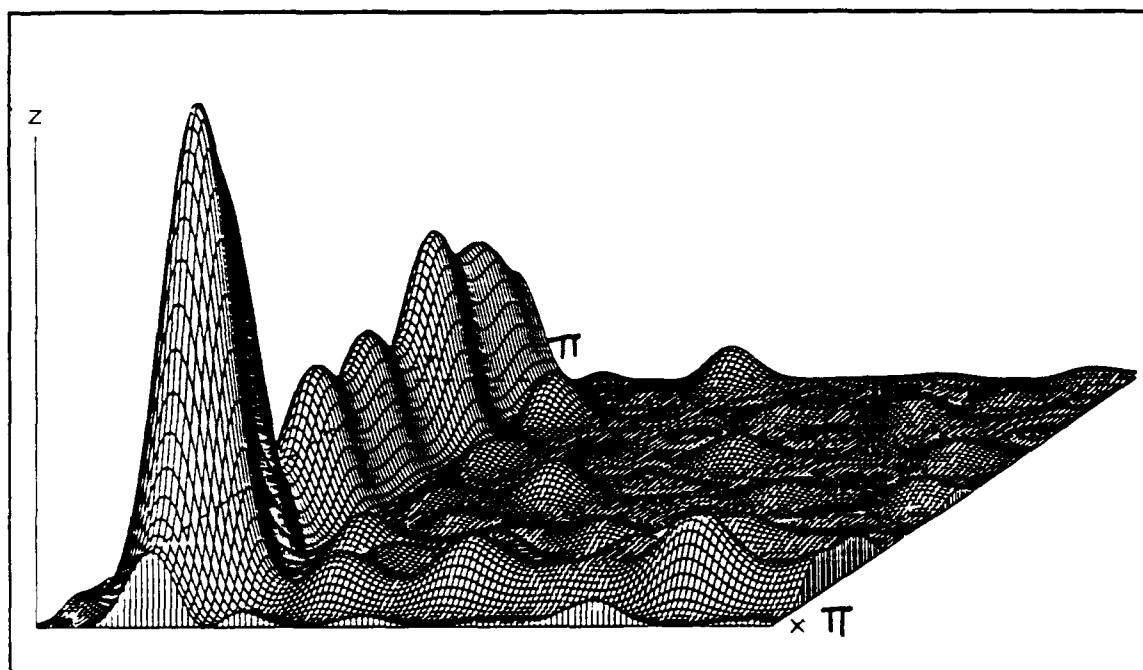


Figure 10: 2-D periodogram for Example 3a using normalized interpolating function.

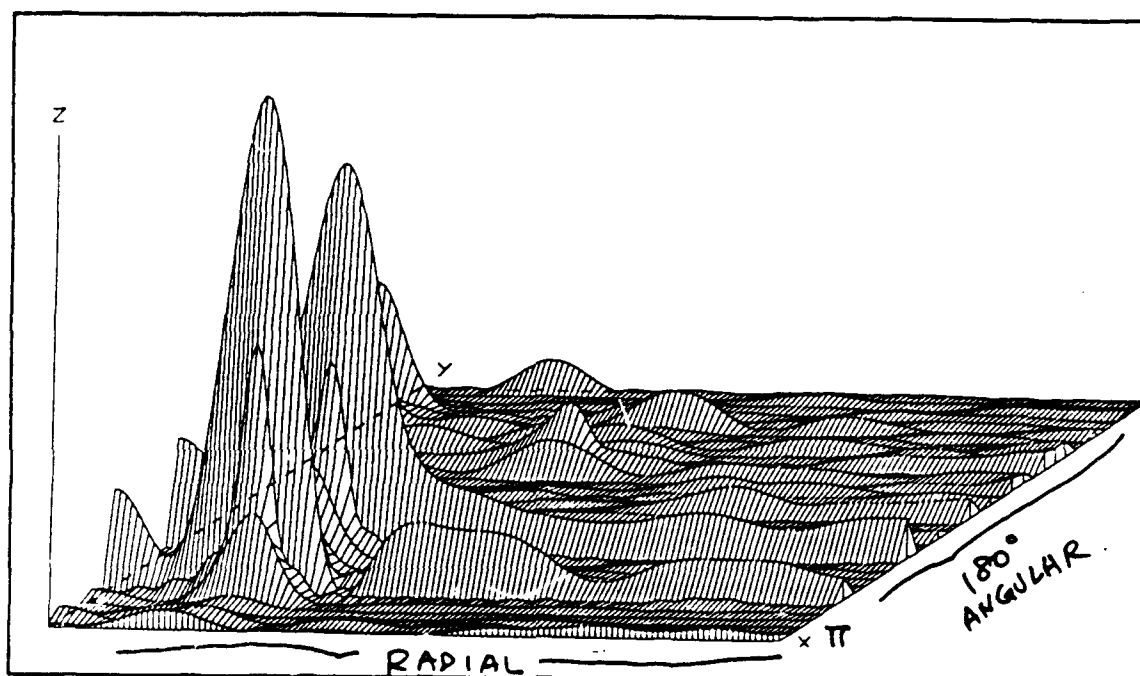


Figure 11: Spectrum for Example 3a using normalized interpolating function and the proposed method.

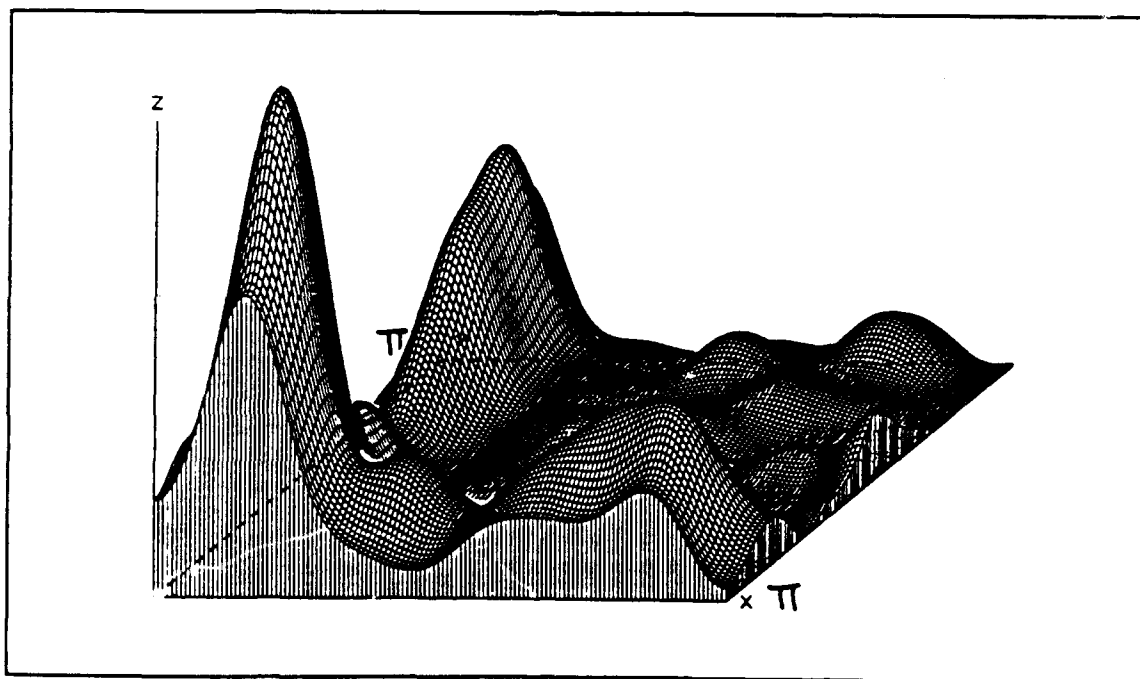


Figure 12: 2-D periodogram for Example 3b using normalized interpolating function.

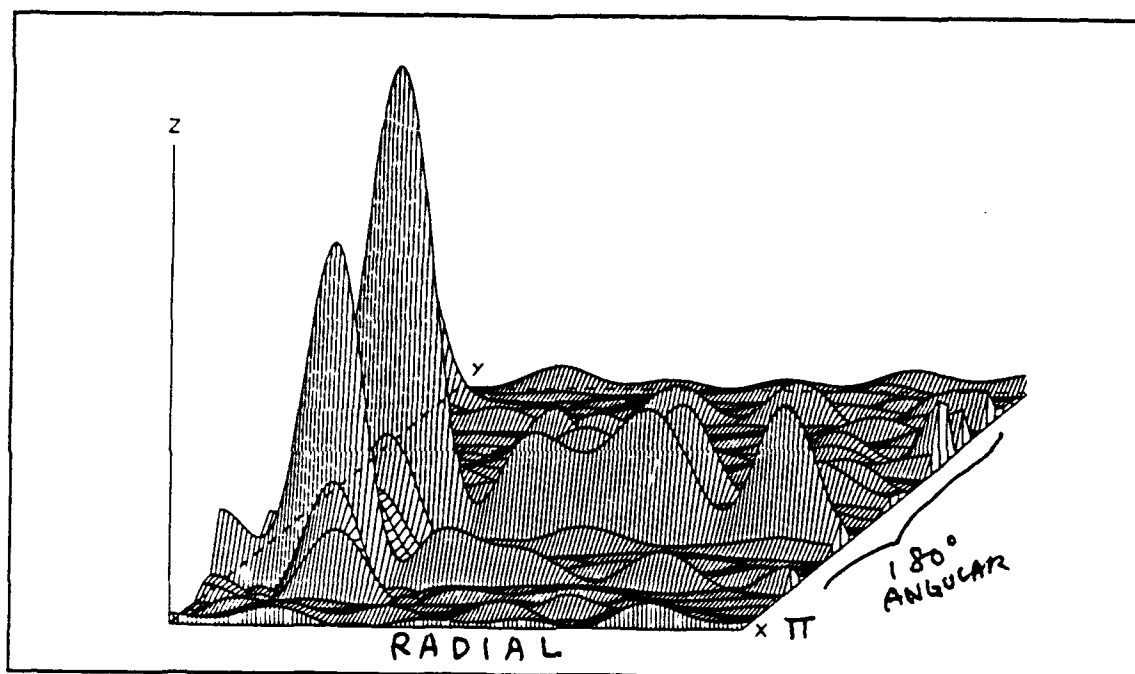


Figure 13: Spectrum for Example 3b using normalized interpolating function and the proposed method.

APPENDIX H

W.-H. Fang and A.E. Yagle, "A Systolic Architecture for New Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," submitted to *IEEE Trans. Signal Processing*.

A Systolic Architecture for New Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices

Wen-Hsien Fang and Andrew E. Yagle
Dept. of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-2122

June 1991

Abstract

Recently, new fast algorithms have been developed for computing the optimal linear least-squares prediction filters for arbitrary Toeplitz-plus-Hankel covariances [1]. In this correspondence, we propose a systolic architecture that can fully express the inherent concurrency of this highly parallelizable algorithm. The simplification of this array structure for centrosymmetric covariances is also addressed.

I INTRODUCTION

The advent of high speed, low cost VLSI devices has changed the field of signal processing dramatically. Due to its tremendous computational capability, more sophisticated algorithms have become feasible through some special-purpose device, e.g. ASICs (application-specific ICs). Under such circumstances, the conventional criterion of number of computations alone is no longer an effective measure of overall performance. The structure of the algorithm and its corresponding hardware architecture play an even more important role. More specifically, an efficient algorithm is defined in terms of its parallelization and the possibility of hardware structures that can fully express its parallelism so that minimal time complexity can be achieved.

Recently, new split algorithms were developed for computing the linear least-squares prediction filters for arbitrary Toeplitz-plus-Hankel covariances [1]. These fast algorithms not only are highly parallel but also perform regular iterative computations. In addition, the Laplacian operator appearing in all the recurrences is an operation involving only closest neighbors. With these desired properties (parallelization and local communication), it is natural that there exist some highly concurrent VLSI computing processors for these fast algorithms such that the overall time complexity can be further decreased. This correspondence confirms this conjecture by proposing some corresponding hardware architectures which are amenable to VLSI implementations.

Special attention will be put on the systolic array architecture. This specific hardware structure (array processors) has several desirable features, such as making multiple use of input data (pipeline processing), using extensive concurrency, involving only a few types of simple cells (saving design cost), and simple and regular data flow (local communication) [2]. To follow we will follow the procedures proposed in [3] to map the fast algorithms of [1] onto some systolic architectures. After we put in the initial conditions, the results will rhythmically pump out of these array processors.

This correspondence is organized as follows. We begin with a brief review of the fast algorithms of [1]. A systolic architecture is then developed to implement these fast algorithms. The array structure

and the required control program are discussed. Its simplification for centrosymmetric covariances will also be addressed. Finally, we conclude this paper with a summary and future perspective.

II SYSTOLIC ARCHITECTURES FOR THE NEW SPLIT ALGORITHMS

A. Review of the New Split Algorithms of [1]

The problem considered is as follows. From the $2i - 1$ noisy observations $\{y_{i-1}, y_{i-2}, \dots, y_{-(i-1)}\}$ of a zero-mean, real-valued discrete random process $\{x_k\}$, compute the linear least-squares estimates of x_i (forward prediction) and x_{-i} (backward prediction) for $i = 1, 1.5, 2, 2.5, \dots$

The observation $\{y_k\}$ are related to the process $\{x_k\}$ by $y_k = x_k + n_k$, where $\{n_k\}$ is a zero-mean discrete-time white noise process with unit power, and $\{x_k\}$ and $\{n_k\}$ are uncorrelated. The estimates of x_i and x_{-i} are computed from the observations using

$$\hat{x}_i = \sum_{j=-(i-1)}^{i-1} h(i, j) y_j; \quad \hat{x}_{-i} = \sum_{j=-(i-1)}^{i-1} h(-i, j) y_j, \quad i = 1, 1.5, 2, \dots \quad (1)$$

The prediction filter $h(i, j)$ are computed by solving the following Wiener-Hopf equation (for $i = \pm 1, \pm 1.5, \pm 2, \dots$)

$$k(i, j) = h(i, j) + \sum_{n=-(i-1)}^{i-1} h(i, n) k(n, j), \quad -(|i| - 1) \leq j \leq |i| - 1 \quad (2)$$

The goal of [1] is to derive fast algorithms for solving (2) when $k(i, j)$ ($\triangleq E[x_i x_j]$) has the Toeplitz-plus-Hankel structure, i.e. $k(i, j) = k_1(i - j) + k_2(i + j)$. For a $(2I_{max} - 1)^{th}$ order linear prediction problem (from $i = -(I_{max} - 1)$ to $I_{max} - 1$), the overall procedure for the new split algorithm of [1] can be summarized as follows:

1. Initializations:

$$\begin{aligned} h(\pm 1, 0) &= -\frac{k(\pm 1, 0)}{1 + k(0, 0)} \\ s(\pm \frac{1}{2}, j) &= k(\pm \frac{1}{2}, j) \quad \text{for } j = \pm \frac{1}{2}, \pm 1\frac{1}{2}, \dots, \pm(2I_{max} - \frac{1}{2}) \\ s(\pm 1, j') &= \delta_{\pm 1, j'} + k(\pm 1, j') - h(\pm 1, 0)k(0, j') \quad \text{for } j' = \pm 1, \pm 2, \dots, \pm(2I_{max} - 1) \end{aligned}$$

2. Computation of the non-local potentials V_i' s:

Computing V_i' s by solving the following 2×2 simultaneous equation:

$$\begin{bmatrix} s(i - \frac{1}{2}, i - \frac{1}{2}) & s(-(i - \frac{1}{2}), i - \frac{1}{2}) \\ s(i - \frac{1}{2}, -(i - \frac{1}{2})) & s(-(i - \frac{1}{2}), -(i - \frac{1}{2})) \end{bmatrix} \begin{bmatrix} V_i^1 \\ V_i^2 \end{bmatrix} = \begin{bmatrix} s(i - \frac{1}{2}, i - \frac{1}{2}) - s(i, i) \\ s(i - \frac{1}{2}, -(i - \frac{1}{2})) - s(i, -i) \end{bmatrix} \quad (3)$$

3. (a) Generalized Levinson algorithm

i. Border points

$$h(i + \frac{1}{2}, i - \frac{1}{2}) = h(i, i - 1) - V_i^1; \quad h(i + \frac{1}{2}, -(i - \frac{1}{2})) = h(i, -(i - 1)) - V_i^2 \quad (4)$$

ii. Nonborder points (for $-(i - \frac{3}{2}) \leq j \leq (i - \frac{3}{2})$)

$$h(i + \frac{1}{2}, j) = h(i, j + \frac{1}{2}) + h(i, j - \frac{1}{2}) - h(i - \frac{1}{2}, j) + V_i^1 h(i - \frac{1}{2}, j) + V_i^2 h(-(i - \frac{1}{2}), j) \quad (5)$$

(b) Generalized Schur algorithm (for $i + \frac{1}{2} \leq j \leq 2I_{max}$)

$$s(i + \frac{1}{2}, j) = s(i, j + \frac{1}{2}) + s(i, j - \frac{1}{2}) - s(i - \frac{1}{2}, j) + V_i^1 s(i - \frac{1}{2}, j) + V_i^2 s(-(i - \frac{1}{2}), j) \quad (6)$$

4. Continue Steps 2 to 3 from $i = 1$ to I_{max} with every step increment $\frac{1}{2}$

where the Schur variable $s(i, j) \triangleq \delta_{i,j} + k(i, j) - h(i, j) - \sum_{n=-(i-1)}^{i-1} h(i, n)k(n, j)$. Note that from (2), $s(i, j) = 0$ if $|j| < |i|$.

B. Systolic Array for the above Fast Algorithm

The above fast algorithms require $24I_{max}^2$ multiplications and divisions, and $48I_{max}^2$ additions and subtractions [1]. To follow, we propose a systolic architecture that can fully exploit the inherent concurrency (parallel and pipeline processing) of this algorithm so that $O(I_{max})$ time complexity can be achieved.

To map this algorithm onto a corresponding array processor, we follow the procedures proposed in [3]. First, a DG (dependence graph) is established in Figure 1, where the shaded regions denote the region of support for the Schur algorithm. A SFG (signal flow graph) can then be derived by mapping

this DG along some feasible direction. In the sequel we choose the mapping along the i direction. Since this is a systolic direction, the resulting SFGs, which are shown in Figures 2 and 3, are also the desired systolic arrays.

Figure 2 shows the array processors with $16I_{max} + 4$ processing elements (PE's) that implement the generalized Schur algorithm, while Figure 3 shows the array processors with $8I_{max} - 6$ PE's that implement the generalized Levinson algorithm. The overall architecture is the combination of both two array processors. Figure 4 shows the operations performed in the right-hand ($i > 0$) upper and lower PE's, respectively. The left-hand ($i < 0$) processing units are the same except that the directions of $V_{\pm i}^{1(2)}$ are reversed. (Note that for clarity, the transmission of $V_{\pm i}^{1(2)}$ are not shown in the array processors of Figures 2 and 3.) For convenience, the array processors in Figures 2 and 3 will be referred to as array S and array L, respectively.

The initial conditions $s(\pm \frac{1}{2}, \pm j)$, $s(\pm 1, \pm j')$, where $j = \frac{1}{2}, 1\frac{1}{2}, \dots, 2I_{max} - \frac{1}{2}$, $j' = 1, 2, \dots, 2I_{max} - 1$, and $h(\pm 1, 0)$ are put in the array S and array L, respectively, before the recursion begins. At first stage of the recursion, the potentials V_i' 's are computed at the four central computing units in array S by using (3), then V_i' 's are sent to all the other processing units in array S to update $s(i, j)$ by using (6), and to array L to update $h(i, j)$ by using (4) for the border points and (5) for the nonborder points.

After completing the updating procedures, the contents in the array S (i.e. $s(i, j)$) are shifted centerward by one unit to prepare for the next recursion. The recursion continues until $i = I_{max}$ with the step of $\frac{1}{2}$ in each recursion. Note that in the updating process, the processing units are activated only on alternate time steps. This is because the updating equations (5) and (6) involve the variables of the previous two time steps. The results of this interleaving update after each time step are shown in Figures 2 and 3. We can find that the variables indexed with integer and half-integer "pop up" alternately. If the computation of the non-local potentials in (3) requires time interval τ_1 and the update plus shifting operations require time interval τ_2 , then the total computing time complexity

would be $(2I_{max} - 2)(\tau_1 + \tau_2)$. Note that since the recurrences perform in-place computations, only $24I_{max} - 2$ memory units are required.

The undesired global transmission (broadcasting) of the non-local potentials V_i' s (see Figure 4) can be avoided by using the concept of computational wavefront proposed in [3], in which the operation performed in each cell is triggered by the availability of the data, instead of by the global clock. The updating processes are finished after the computational wavefront propagates from the center to the right (left) end, for which the computing time becomes $(2I_{max} - 2)(\tau_1 + \tau_2) + I_{max}\tau_2$ by assumption. The extra time $I_{max}\tau_2$ is the price to avoid the global communication scheme.

A program, which adopts the same notations used in [4] and summarized the above procedures, is shown in Figure 5. This control program is broadcast to each PE before the arrays begin the recursion. Note that further simplifications are possible. Since the arrays S and L perform almost the same type of operations with complementary support, we can combine both arrays into a single one with a suitable partition. Also, since the PE's are only active at alternate time step, pairs of adjacent processing units can be combined together so that the number of the PE's can be reduced by one half.

If we solve (2) directly using the Gaussian elimination procedure, $O(I_{max}^3)$ multiplications and divisions, and $O(I_{max}^2)$ memory units are required using a sequential machine. Furthermore, this is not a highly parallelizable procedure. Merchant and Parks provided an efficient alternative to compute the Toeplitz-plus-Hankel coefficient matrix system of equations [5]. However, their approach is to reformulate the original system into a block-Toeplitz system, and then solve it by the multichannel Levinson algorithm, which not only requires much more complex computations (e.g. matrix inversion), but also needs larger data bus and more memory space.

C. Simplification of the Array Structure for Centrosymmetric Covariances

In the special case that $k(i, j) = k(-i, -j)$, i.e. a centrosymmetric covariance matrix, we have $h(i, j) = h(-i, -j)$, $s(i, j) = s(-i, -j)$, $V_i^1 = V_{-i}^2$, $V_i^2 = V_{-i}^1$ [1]. Hence the arrays for $i < 0$ can be dispensed with.

Further simplification is possible if we define

$$a(i, j) \triangleq h(i, j) + h(i, -j), \quad e(i, j) \triangleq s(i, j) + s(i, -j), \quad V_i \triangleq V_i^1 + V_i^2 \quad (7)$$

then we can get the recursive expressions for $a(i, j)$ and $e(i, j)$, respectively, as follows

$$a(i + \frac{1}{2}, j) = a(i, j + \frac{1}{2}) + a(i, j - \frac{1}{2}) - a(i - \frac{1}{2}, j) + V_i a(i - \frac{1}{2}, j) \quad (8)$$

$$e(i + \frac{1}{2}, j) = e(i, j + \frac{1}{2}) + e(i, j - \frac{1}{2}) - e(i - \frac{1}{2}, j) + V_i e(i - \frac{1}{2}, j) \quad (9)$$

and the new non-local potential V_i can be computed by

$$V_i = [e(i - \frac{1}{2}, i - \frac{1}{2}) - e(i, i)] / e(i - \frac{1}{2}, i - \frac{1}{2}) \quad (10)$$

Similarly, we can define

$$a^*(i, j) \triangleq h(i, j) - h(i, -j), \quad e^*(i, j) \triangleq s(i, j) - s(i, -j), \quad V_i^* \triangleq V_i^1 - V_i^2 \quad (11)$$

and we can get the same recurrences for $a^*(i, j)$, $e^*(i, j)$ and V_i^* as (8), (9), and (10), respectively.

The array processors for solving the centrosymmetric matrix systems are shown in Figure 6, where four array processors are constructed to update $a(i, j)$, $a^*(i, j)$, $e(i, j)$, and $e^*(i, j)$ respectively. The operations performed in each PE are similar to those of Figure 4. The division cells (DIV) are used to compute the non-local potentials V_i (V_i^*), which are then used to update $a(i, j)$ ($a^*(i, j)$) and $e(i, j)$ ($e^*(i, j)$), respectively. The resulting $h(i, j)$ and $h(i, -j)$ can be derived by

$$h(i, j) = \frac{a(i, j) + a^*(i, j)}{2}; \quad h(i, -j) = \frac{a(i, j) - a^*(i, j)}{2} \quad (12)$$

In Figure 6, $2(2I_{max} - 1)$ PE's are required for $e(i, j)$ and $e^*(i, j)$, and $2I_{max}$ PE's are required for $a(i, j)$ and $a^*(i, j)$. Note that here we put two adjacent points $((i, j)$ and $(i - \frac{1}{2}, j - \frac{1}{2}))$ in each PE, so the overall memory required is $4(2I_{max} - 1) + 4I_{max} = 12I_{max} - 4$ units. If we use the complementary support property of $e(i, j)$ ($e^*(i, j)$) and $a(i, j)$ ($a^*(i, j)$), then we can put $a(i, j)$ ($a^*(i, j)$) at the end of arrays $e(i, j)$ ($e^*(i, j)$) and use only $2(2I_{max} - 1) + 1$ PE's plus $4(2I_{max} - 1) + 2$ memory units.

If the division requires time interval τ'_1 and the update plus shifting operations require time interval τ_2 , then the total computing time complexity would be $(2I_{max} - 2)(\tau'_1 + \tau_2)$. Again, if we use the data-driven computational wavefront, then the computing time becomes $(2I_{max} - 2)(\tau'_1 + \tau_2) + I_{max}\tau_2$ by assumption.

Since the symmetric Toeplitz matrix is a special case of a centrosymmetric matrix, we can compare this array architecture with those proposed for solving the Toeplitz system of equations [6, 7]. We find that not only is the architecture simpler, but also the overall computational time is reduced. This is not surprising because we are concerned with a linear prediction problem which has specific right-hand side in the matrix equation, instead of solving a general Toeplitz system of equations, which requires the inversion of a Toeplitz matrix followed by a back substitution operation. Applying our proposed architecture to arbitrary centrosymmetric systems of equations would require additional processors for the back substitution. Nevertheless, the proposed architecture is capable of solving more general problems (applicable to *arbitrary* Toeplitz-plus-Hankel or centrosymmetric covariances) than those of [6, 7].

III CONCLUSION

In this correspondence, we have developed a systolic architecture to implement the recently-developed fast algorithms of [1] to compute the optimal linear least-squares prediction filters for arbitrary Toeplitz-plus-Hankel covariances. The overall time complexity for computing the $(2I_{max} - 1)^{th}$ order linear prediction filters is reduced from $O(I_{max}^2)$ to $O(I_{max})$ by using only $O(I_{max})$ PE's and $O(I_{max})$ storage. Some issues that need further research are as follows. Modifications of the above systolic architecture so that it is capable of solving more general Toeplitz-plus-Hankel coefficient matrix system of equations. Extension of this architecture to the 2-D counterpart [8] of the above 1-D fast algorithms.

ACKNOWLEDGMENT

This work was supported by the Air Force Office of Scientific Research under grant # AFOSR-89-0017. The first author would like to thank Mr. Tsunglun Yu for many useful comments and discussions.

References

- [1] A. E. Yagle, *New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices*, to appear in IEEE Trans. Sig. Proc., Nov. 1991.
- [2] H. T. Kung, *Why Systolic Array?*, Computer, Vol. 15, pp. 37-46, Jan. 1982.
- [3] S. Y. Kung, *VLSI Array Processors*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- [4] J. D. Ullman, *Computational Aspects of VLSI*, Computer Science Press, Rockville, MD, 1984.
- [5] G. A. Merchant and T. W. Parks, *Efficient Solution of a Toeplitz-plus-Hankel Coefficient Matrix System of Equations*, IEEE Trans. Acoust., Speech, Sig. Proc., Vol. ASSP-30, No.1, pp. 40-44, Feb. 1982.
- [6] S. Y. Kung and Y. H. Hu, *A Highly Concurrent Algorithm and Pipelined Architecture for Solving Toeplitz Systems*, IEEE Trans. Acoust., Speech, Sig. Proc., Vol. ASSP-31, No.2, pp. 66-75, Feb. 1983.
- [7] R. P. Brent and F. T. Luk, *A Systolic Array for the Real-Time Solution of Toeplitz Systems of Equations*, J. VLSI Comput. Syst., Vol. 1, pp. 1-22, 1983.
- [8] W.-H. Fang, A.E. Yagle, *Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster*, to appear in IEEE Trans. Sig. Proc., June 1992.

FIGURE HEADING

1. Figure 1: Dependence graph for the 1-D generalized Levinson and Schur algorithms.
2. Figure 2: Systolic architecture for the 1-D generalized Schur algorithm.
3. Figure 3: Systolic architecture for the 1-D generalized Levinson algorithm.
4. Figure 4: The operations performed in the right-hand upper PE's (left) and lower PE's (right) (except the boundary PE's of array L).
5. Figure 5: Program performs the update procedures in each PE.
6. Figure 6: Systolic architecture for solving the centrosymmetric matrix system

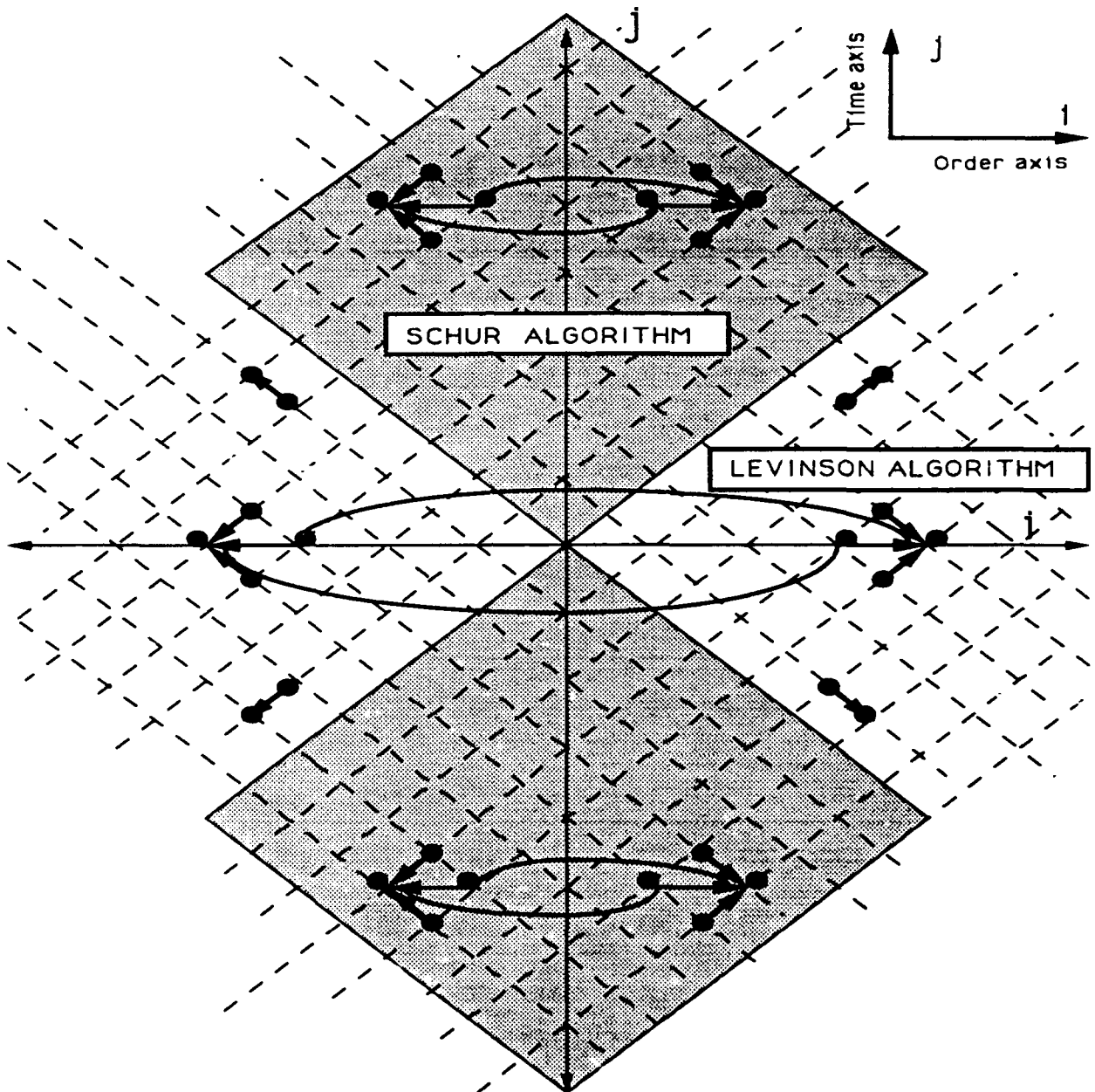


Figure 1: Dependence graph for the 1-D generalized Levinson and Schur algorithms

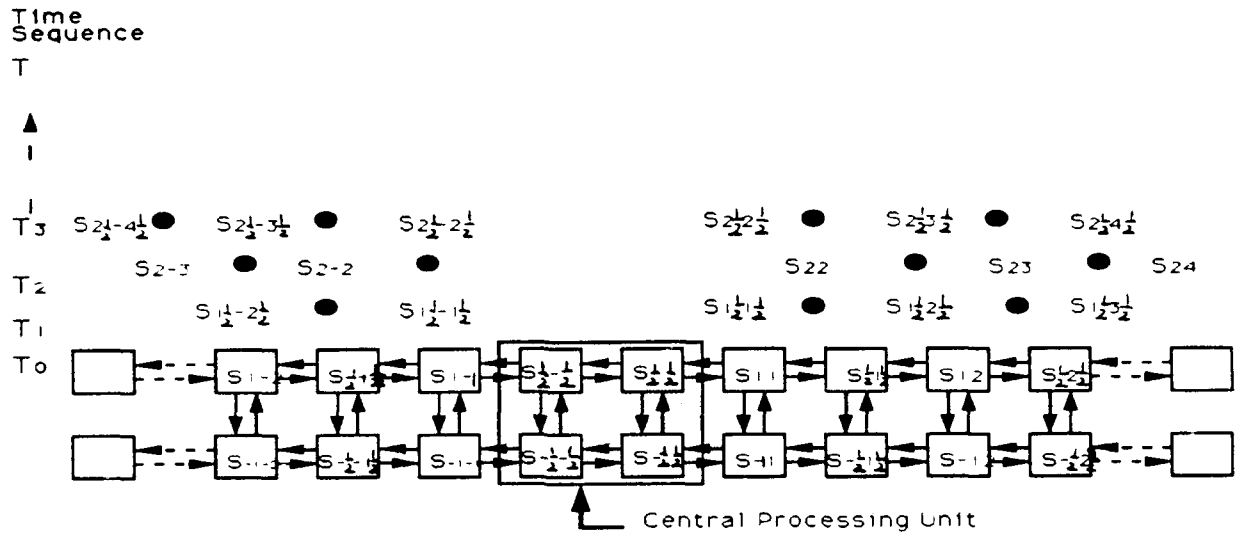


Figure 2: Systolic architecture for the 1-D generalized Schur algorithm

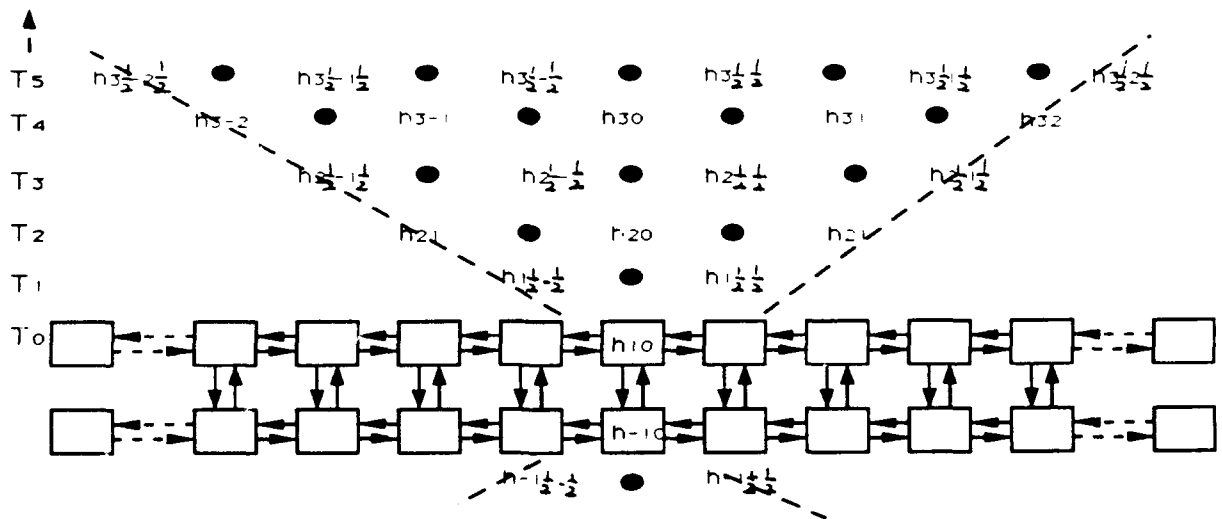


Figure 3: Systolic architecture for the 1-D generalized Levinson algorithm

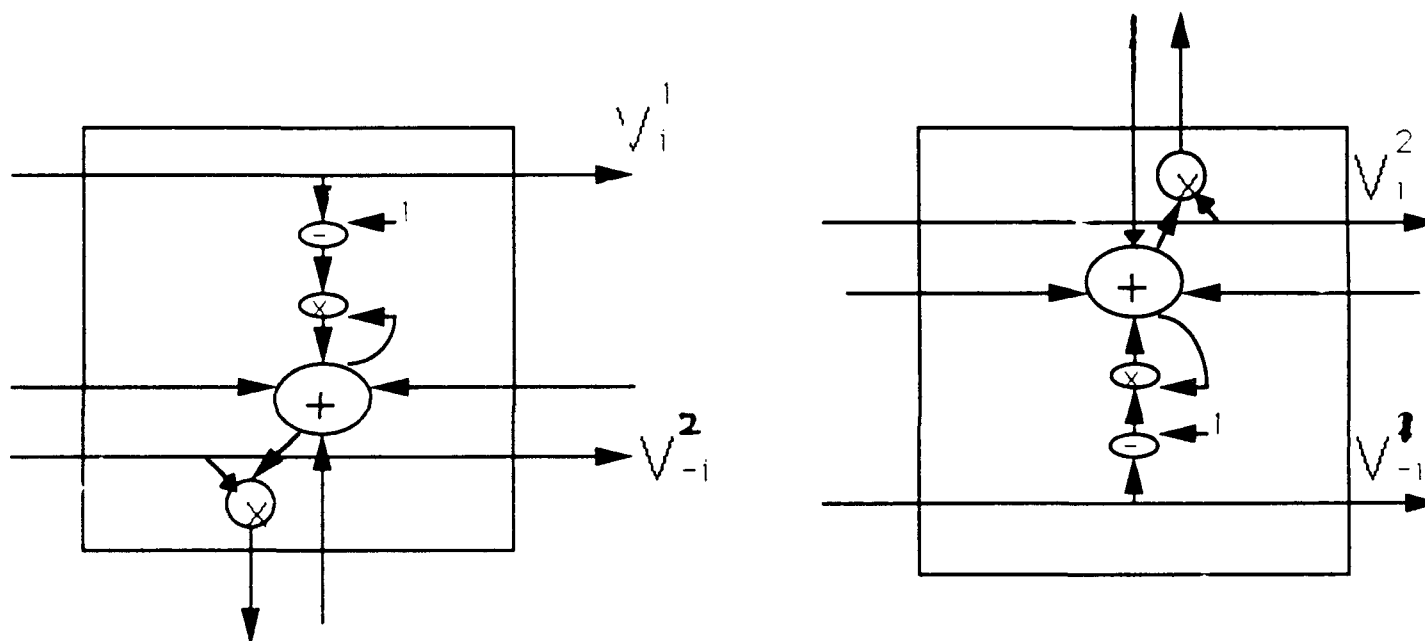


Figure 4: The operations performed in the right-hand upper PE's (left) and lower PE's (right) (except the boundary PE's of array L)

```

Program { for PE  $j$  and the prediction filter at point  $i + \frac{1}{2}$  }
    {  $i$  will increase by  $\frac{1}{2}$  at the end of each recursion }
    for  $j := -(i - \frac{1}{2})$  to  $(i - \frac{1}{2})$  do beat begin {  $i$  is an integer }
        receive  $V_i$  from left(right) neighbor;
        send  $V_i$  to right(left) neighbor;
        { transmission of the non-local potential }
        if  $j$  is a half integer, then begin
            if PE is non-border cell (  $j \neq \pm(i - \frac{1}{2})$  ) then do equation (5);
            else do equation (4);
            end;
        else {  $j$  is an integer }
            PE do nothing;
        beat end;
    beat begin {  $j$  is an integer } {  $i$  is a half-integer }
    { same procedures by switching the role between "integer" and "half-integer" }
    beat end;
end;

```

Figure 5: Program performs the update procedures in each PE

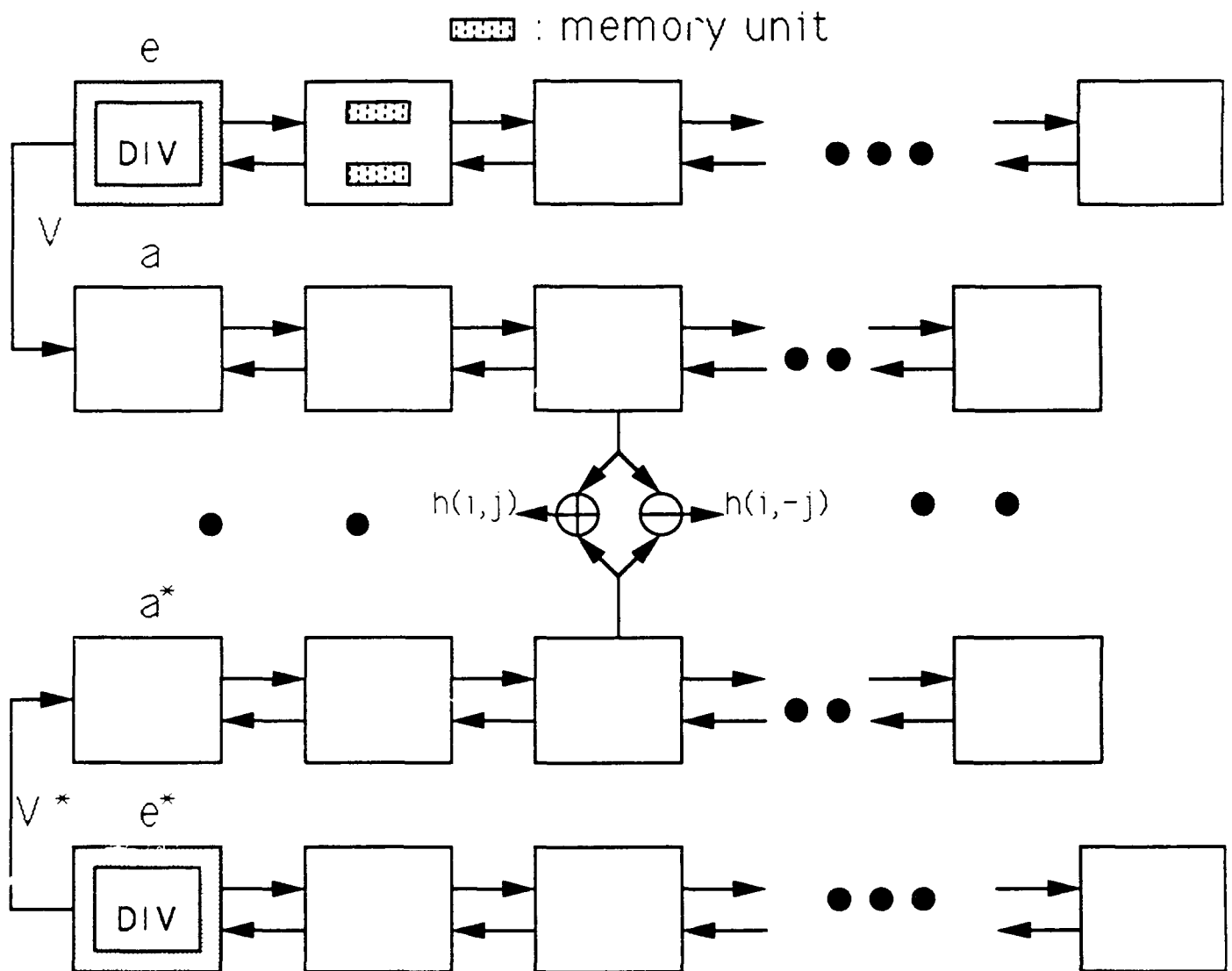


Figure 6: Systolic architecture for solving the centrosymmetric matrix system

APPENDIX I

Experimental Results

In this section, we provide some simple simulation results by applying the fast algorithm developed above to the image restoration (smoothing) and coding problems.

1 Image Restoration and Smoothing

The objective of the image restoration and smoothing is to recover the original image from a degraded one which is contaminated by some sort of noise. Here, we consider the most common case that the noise is the additive white noise and the method employed to reduce the observation noise is the linear least squares prediction or smoothing.

The comparison criterion is the improvements of the Signal-to-Noise ratio (ISNR), which is defined as

$$\begin{aligned} ISNR (dB) &= 10 \log \frac{\text{average signal power}}{\text{average power of prediction error}} - 10 \log \frac{\text{average signal power}}{\text{average power of observation error}} \\ &= 10 \log \frac{\text{average power of observation error}}{\text{average power of prediction error}} \end{aligned}$$

For each set of data, four types of algorithms are used to compute the resulting ISNR. These four algorithms include : Linear Prediction (LP), Linear Prediction on zero mean residues (LPZM), Smoothing (SM), and Smoothing on zero mean residues (SMZM). LP is to use the fast algorithm developed to compute the linear prediction filter, and SM is to compute the smoothing filter by combining the LP and the BSK identity. LPZM (SMZM) means that the linear prediction (smoothing) filter is applied on the zero mean residues which are derived by subtracting the global mean from the original signals. For simplicity, the observation noise is the white noise with unit power. The prediction coefficients are generated by assuming that the covariance function has the form as ρ^{-r} ($\rho = 0.995 \approx 1$ and r is the distance from the origin) so that the requirement of the covariance having Toeplitz-plus-Hankel structure is satisfied.

From figures (1) to (4), four different isotropic random fields are generated. The covariance functions for these four isotropic random fields are $4(0.82)^r$ for figure (1), $7(0.78)^r$ for figure (2),

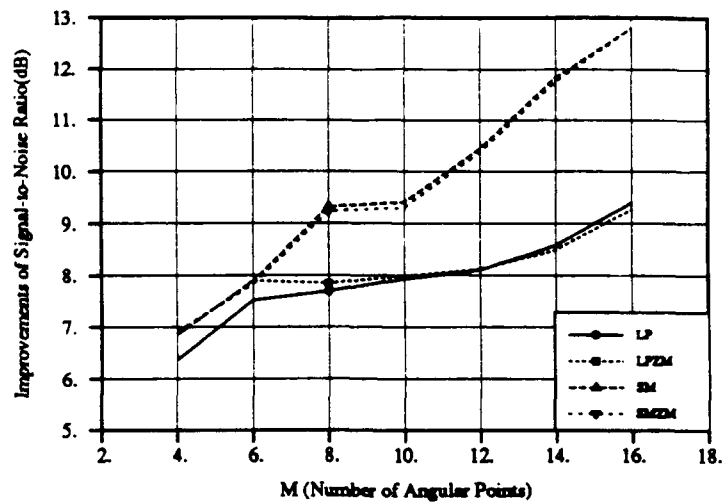


Figure 1: The Comparison of ISNR for Different M with Covariance Function $4(0.82)^r$

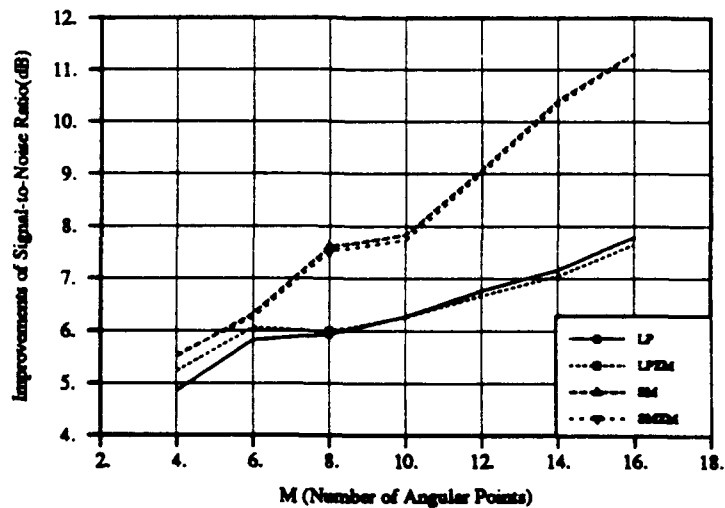


Figure 2: The Comparison of ISNR for Different M with Covariance Function $7(0.78)^r$

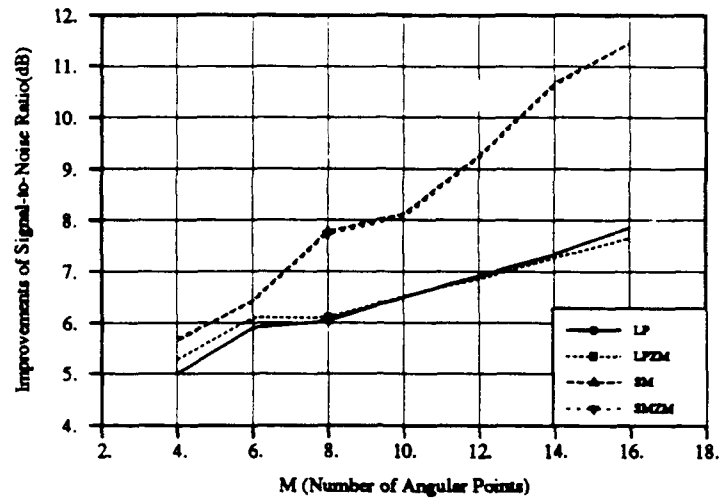


Figure 3: The Comparison of ISNR for Different M with Covariance Function $12\frac{0.78r}{r}$

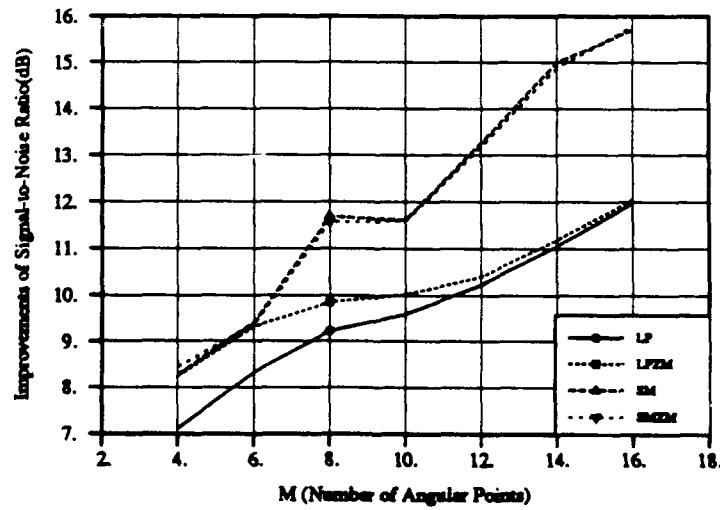


Figure 4: The Comparison of ISNR for Different M with Covariance Function of $3.5K_1(r)$ Form

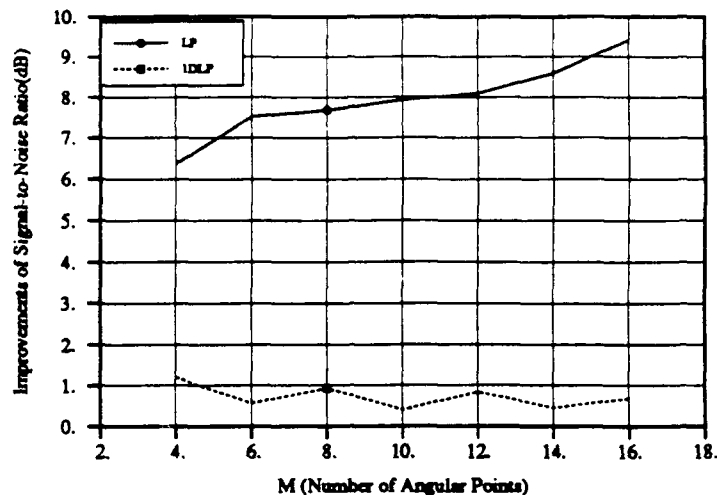


Figure 5: The Comparison of ISNR for 1DLP and LP for Different M

$12\frac{0.78r}{r}$ for figure (3) , and $3.5K_1(r)$ for figure (4) respectively. The points along each direction are fixed to 10, and ISNR is computed for different M (number of angular points).

The simulation results show that in general ISNR improves as M increases. That's a reasonable result since with more data points (information) available, we can get a more accurate prediction of the original signal. The same argument can also be applicable to the result that ISNR using SM is larger than that using LP. The latter is furtherly supported by the results that the difference of ISNR for LP and SM becomes larger as M increases, which reflects the fact that the data points available for SM are propotional to M so that the the difference of the data being available increase as M increases.

The ISNR for LPZM (SMZM) are slightly better than that for LP(SM) in small M and are approximately the same for larger M . This may be explained that when only small amount of data are available, LPZM (SMZM) satisfy the zero mean assumption and produce a more accurate prediction (smoothing). But as more data are available, the data generated will be approximate

zero mean so that both results won't make much difference.

It's worth noting that the simulation results in figures (3) and (4) are similar to those of the previous figures regardless of the mismatch of the covariance function. This striking result shows that even the linear prediction (smoothing) filters are generated by the wrong assumption of the covariance functions, the resulting ISNR is still satisfactory as long as the random field is isotropic and highly correlated, which happens quite often in the practical images. The ISNR in figure (4) is better than that in figure (3) because the covariance function in figure (4) is more correlated and does not decay as fast as that of figure (3). This highly correlated covariance, *i.e.* $\rho \approx 1$, is the requirement to derive the above fast algorithm.

The results of the LP which use *all* the available data on a polar raster are better than those use only the data on the same line, which is equivalent to 1-D linear prediction problem (1DLP). As shown in the figure (5), the ISNR for LP is always larger than that for 1DLP. In addition, since the linear prediction only utilizes the previous sample in the 1DLP, the ISNR will be approximately the same for all M , which is opposed to that for LP (SM). This is another advantage of LP over 1DLP. Although the algorithm for the latter is faster by using the 1-D Levinson algorithm, however, the performance is worse.

These simulation results confirm our claim that these two algorithms (for prediction and smoothing) work well independent of the value of M , although the performance gets better as M increases, *i.e.* more data are available.

2 Linear Predictive Coding of Images

We can note that the linear predictive coefficients can be obtained as long as the covariance function is available. Therefore, we can either store or transmit the residues of the data instead of the data itself and accompany the covariance function as the side information. Since the residues are derived by subtracting the linear combination of the previous data from the present data to reduce the unnecessary redundancy, hence they are in general smaller than the data themselves. Besides, in many cases only few parameters, e.g. ρ in the isotropic random field, would be required to specify the covariance function. Therefore, the overall storage requirement can be reduced significantly in the finite precision environment.

We take the previous data as examples by considering the noisy images as the original image and the prediction errors as the prediction residues. The data in tables (1) and (2) are the same as those of figures (4) and (7) respectively. In the following tables, we compare the average signal power and the resulting prediction residues using both the LP and 1DLP.

The experiments show that the results using LP always provide the optimal performance and are significantly smaller than the average signal power, thereby the storage requirements can be reduced. It must be emphasized that the performance depends on the test images. The results get worse when there are large variations in the images large, e.g. edges or lines. This is the limitation of the batch least-square method which takes into account of all the data, so that the results can not adapt the quick change of the outside environment. The above result can also be regarded as a tradeoff between performance and complexity. Although complicated algorithms would take lots of time, it would also provide optimal performance, i.e. require minimal storage requirements.

M	Average Signal Power (dB)	1DLP (dB)	LP (dB)
4	3.73	-1.04	-6.22
6	3.69	-1.16	-8.16
8	3.70	-1.55	-8.32
10	3.69	-0.52	-8.56
12	3.68	-1.24	-8.59
14	3.68	-0.88	-9.04
16	3.68	-1.15	-9.89

Table 1: Comparison of Average Signal and Residues Power Using 1DLP and LP for Different M
(Number of Angular Points)

M	Average Signal Power (dB)	1DLP (dB)	LP (dB)
4	5.69	-1.10	-6.97
6	5.68	-1.04	-8.95
8	5.68	-1.57	-9.84
10	5.68	-1.36	-10.12
12	5.68	-1.29	-10.22
14	5.67	-0.97	-11.07
16	5.67	-1.16	-12.03

Table 2: Comparison of Average Signal and Residues Power Using 1DLP and LP for Different M
(Number of Angular Points)

APPENDIX J

1. A.E. Yagle, "New Analogues of Split Algorithms for Toeplitz-plus-Hankel Matrices," 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing, Toronto, Canada, May 14-17, 1991, pp. 2253-2256.
2. W.-H. Fang and A.E. Yagle, "Discrete Fast Algorithms for Two-Dimensional Linear Prediction on a Polar Raster," 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, NM, April 3-6, 1990, pp. 2017-2020.
3. A.E. Yagle, "Generalized Levinson and Fast Cholesky Algorithms for Three Dimensional Random Field Estimation Problems," Sixth Multidimensional Signal Processing Workshop, Asilomar, Pacific Grove, CA, September 6-8, 1989. p. 145.

NEW ANALOGUES OF SPLIT ALGORITHMS FOR TOEPLITZ-PLUS-HANKEL MATRICES

Andrew E. Yagle

Dept. of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, Michigan 48109

ABSTRACT

New fast algorithms for solving arbitrary Toeplitz-plus-Hankel systems of equations are presented. The algorithms are analogues of the split Levinson and Schur algorithms, although the more general Toeplitz-plus-Hankel structure requires that the algorithms be based on a four term recurrence: relations with previous split algorithms are noted. The algorithms require roughly half as many multiplications as previous fast algorithms for Toeplitz-plus-Hankel systems.

I. INTRODUCTION

Toeplitz-plus-Hankel (TH) systems of equations have many important applications, such as linear prediction for nonstationary processes with TH covariances, two-sided autoregressive spectral estimation [1], linear-phase prediction filter design [2], Hildebrand-Prony spectral line estimation procedure [3], and PADE approximation to the cosine series expansion of an even function [4]. Integral equations with a TH kernel arises in atmospheric scattering [5] and rarefied gas dynamics [6].

Fast algorithms for TH systems have appeared in [7]-[9]. The new algorithms of this paper can be viewed as split versions of those of [8], or as generalizations of the split algorithms of [10] from symmetric Toeplitz to arbitrary TH systems.

The heart of the new algorithms is a four-term recurrence that generalizes the three-term recurrences of [10] to TH matrices. This recurrence requires two multiplications per update, half the number required by the algorithms of [7]-[9]. This is analogous to the 50% savings in multiplications for the split algorithms of [10] over the classical Levinson and Schur algorithms.

II. DERIVATION OF FOUR-TERM RECURRENCE

A. The Basic Problem

We consider the solution of the TH system

$$\begin{bmatrix} 1+k_{-i,-1} & \dots & k_{-i,0} \\ \vdots & \ddots & \vdots \\ k_{-1,0} & \dots & k_{-1,i} \\ \vdots & \ddots & \vdots \\ k_{-1,i} & \dots & 1+k_{-1,i} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ h_{-i,-(i-1)} & h_{-i,-(i-1)} \\ \vdots & \vdots \\ h_{-1,-1} & h_{-1,-1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} S_{-i,-1} & S_{-i,-1} \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ S_{-1,i} & S_{-1,i} \end{bmatrix} \quad (1)$$

where the $S_{\pm i, \pm i}$ are defined from the $\{k_{i,j}\}$ and $\{h_{i,j}\}$ in (15) below, and the ij^{th} element of the system matrix has the form

$$k_{i,j} = k_1(i-j) + k_2(i+j) \quad (2)$$

for arbitrary functions $k_1(\cdot)$ and $k_2(\cdot)$. Note in particular that the system matrix need be neither symmetric nor persymmetric; the only requirement is that all of the central submatrices be nonsingular.

Updating (1) from i to $i+1$ increases the size of the matrix by two; this requires two updates, and requires $k_{i/2, j/2}$ be defined at half-integer values $(i/2, j/2)$. If $i/2 + j/2$ is not an integer, let $k_{i/2, j/2} = 0$; if $i/2 + j/2$ is an integer, assign $k_{i/2, j/2}$ such that the matrix with i^{th} coordinate $k_{i/2, j/2}$ is TH. If $k_{i,j}$ is specified by the form (2), this can be done easily by inserting the half-integer values in the functions $k_1(\cdot)$ and $k_2(\cdot)$ (note that the arguments will always be integers).

Omitting the first and last rows of (1) allows it to be rewritten as

$$0 = k_{i,j} + h_{i,j} + \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,j}, \quad -(i-1) \leq j \leq i-1. \quad (3)$$

Now define the interpolated system of (3) as

$$0 = k_{i+1/2, j+1/2} + h_{i+1/2, j+1/2} + \sum_{n=-(i-1/2)}^{i-1/2} h_{i+1/2, n} k_{n, j+1/2} \quad (4)$$

and similarly for $-i-1/2$. The interpolated systems for various orders are auxiliary systems of TH systems that are solved along with (3) by the algorithms to follow. This artifice is necessary in order to obtain split algorithms solving nested systems.

B. Derivation of Four-Term Recurrence for $h_{i,j}$

To make the derivation easier to follow, we consider only positive i . Define the discrete wave operator Δ of a function $f_{i,j}$ as

$$\Delta f_{i,j} = f_{i+1/2, j} + f_{i-1/2, j} - f_{i, j+1/2} - f_{i, j-1/2} \quad (5)$$

Δ is the discrete version of the continuous operator $(\frac{\partial^2}{\partial i^2} - \frac{\partial^2}{\partial j^2})$. Note that the TH structure (2) is equivalent to

$$\Delta k_{i,j} = 0; \text{ for integer } i + j. \quad (6)$$

Apply the operator Δ to (3) by writing (3) with i replaced with $i \pm 1/2$, and then j replaced with $j \pm 1/2$, and then adding and subtracting (4) appropriately. Using (5), (6), the non-singularity of (1), and some algebra gives [16]

$$\Delta h_{i,j} = V_i^{-1} h_{i-1/2,j} + V_i^{-2} h_{-(i-1/2),j}, \quad -(i-3/2) \leq j \leq i-3/2 \quad (7)$$

where we have defined the potentials [11]

$$V_i^{-1} = h_{i+1/2,i-1/2} - h_{i,i-1}; \quad V_i^{-2} = h_{i+1/2,-(i-1/2)} - h_{i,-(i-1)}. \quad (8)$$

Eq. (7) can be written as

$$h_{i+1/2,j} = h_{i,j+1/2} + h_{i,j-1/2} + (V_i^{-1} - 1)h_{i-1/2,j} + V_i^{-2}h_{-(i-1/2),j} \quad (9)$$

This is the four-term recurrence at the heart of the new algorithms. It is analogous to the three-term recurrence on which the split algorithms of [10] are based, although there are some differences (see Section VI).

III. NEW SPLIT LEVINSON ALGORITHM

The four-term recurrence (9) can be propagated in increasing $|i|$ and $-(|i| - 3/2) \leq j \leq |i| - 3/2$. Note that for i an integer/half-integer, j will take half-integer/integer values, respectively. However, since (9) does not hold for $j = \pm(i-1)$, we must update $h_{i,\pm(i-1)}$ using (8), and similarly for $h_{-i,\pm(i-1)}$. Also, (8) and (9) require V_i^{-1} and V_i^{-2} to be supplied separately, computed from $k_{i,j}$; note that (8) cannot be used to compute V_i^{-1} and V_i^{-2} , since (8) is needed to update $h_{\pm i,\pm(i-1)}$. We now show how V_i^{-1} and V_i^{-2} can be computed from previously computed $h_{i,j}$ and $k_{i,j}$.

A. Computation of V_i^{-1} and V_i^{-2}

Setting $j = i - 1$ in (3) and (4) gives

$$h_{i+1/2,i-1/2} = -k_{i+1/2,i-1/2} - \sum_{n=-(i-1/2)}^{i-1/2} h_{i+1/2,n} k_{n,i-1/2} \quad (10a)$$

$$h_{i,i-1} = -k_{i,i-1} - \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,i-1}. \quad (10b)$$

Eq. (10b) requires only $k_{i,j}$ (known) and $h_{i,j}$ (from the previous recursion); however, (10a) requires $h_{i+1/2,j}$, which has not yet been computed. Substituting (19) into (10a) and much algebra results in the following. Define the Schur variables

$$S_{i,j} = \delta_{i,j} + k_{i,j} + \sum_{n=-(i-1)}^{i-1} h_{i,n} k_{n,j}, \quad j = \pm i. \quad (11)$$

Note $S_{i,j}$ can be computed from known $k_{i,j}$ and $h_{i,j}$. Then it may be shown that

$$\begin{bmatrix} S_{i-1/2,i-1/2} & S_{-(i-1/2),i-1/2} \\ S_{i-1/2,-(i-1/2)} & S_{-(i-1/2),-(i-1/2)} \end{bmatrix} \begin{bmatrix} V_i^{-1} \\ V_i^{-2} \end{bmatrix} = \begin{bmatrix} S_{i-1/2,i-1/2} - S_{i,i} \\ S_{i-1/2,-(i-1/2)} - S_{i,-i} \end{bmatrix} \quad (12)$$

The existence of a unique solution to (12), which can easily be found in closed form, is proved in Section V below.

B. New Split Levinson Algorithm

Initialization: $h_{\pm 1,0} = -k_{\pm 1,0}/(1 + k_{0,0})$

Computation of V_i^{-1} , V_i^{-2} : Compute $S_{i,\pm i}$ from $k_{i,j}$ (known) and $h_{i,j}$ (from previous recursion) using (11). Compute V_i^{-1} and V_i^{-2} from $S_{i,\pm i}$ and $S_{i-1,\pm(i-1)}$ using (12).

Update $h_{i,j}$: Compute $h_{\pm(i+1/2),\pm(i-1/2)}$ using (8).

Compute $h_{i+1/2,j}$, $|j| \leq (i-3/2)$ using (9).

Compute $h_{-(i+1),j}$ similarly using (7).

At this point the recursion is complete. The computed $h_{i,j}$ for integer/half-integer i and j solve the original system (3)/interpolated system (4), respectively; note that two recursions are needed to increase the size of the system (3) by two (i.e., update i to $i+1$).

This algorithm differs from the split Levinson algorithm of [10] in two respects. First, the non-symmetric TH system matrix requires four sequences V_{\pm}^{-1} and V_{\pm}^{-2} of potentials and the four-term recurrence (13). The symmetric Toeplitz system matrix solved by the split Levinson algorithm of [10] requires only one sequence of potentials and a three-term recurrence. Second, the split Levinson algorithm of [10] propagates not $h_{i,j}$ but $h_{i,j} + h_{i,-j}$; this is more efficient for symmetric Toeplitz matrices, but requires recovery of $h_{i,j}$ from $h_{i,j} + h_{i,-j}$ at termination.

IV. NEW SPLIT SCHUR ALGORITHM

The "inner product" (11) is a computational bottleneck, as in the classical Levinson algorithm. We now derive a new split Schur-type algorithm for arbitrary TH matrices. This algorithm can be propagated in parallel with the split Levinson algorithm derived above: this avoids the computational bottleneck (11). The same idea was used for the classical Schur and Levinson algorithms in [12].

The first step is to show that the forward prediction error filter satisfies the four-term recurrence (9). From this, we show that the $S_{i,j}$ defined in (11) (now for all $j \geq i$) also satisfy (9). Then (9), initialized using $k_{i,j}$, can be used to compute V_i^{-1} and V_i^{-2} quickly.

A. Four-Term Recurrence for $S_{i,j}$

Define $\phi_{i,j}$ as the forward prediction error filter $\phi_{i,j} = \delta_{i,j} + h_{i,j}$. Clearly $\phi_{i,j}$ satisfies (9) for $-(i-3/2) \leq j \leq i-3/2$ since $\phi_{i,j} = h_{i,j}$ for these values. At $j = \pm(i-1/2)$ or $\pm(i+1/2)$ $\phi_{i,j}$ satisfies (9), since this reduces to (8). And for $|j| \geq i+3/2$ (9) reduces to $0 = 0$. Hence (9) with $h_{i,j}$ replaced with $\phi_{i,j}$ is true for all i an integer/half-integer and j a half-integer/integer:

$$\phi_{i+1/2,j} = \phi_{i,j+1/2} + \phi_{i,j-1/2} + (V_i^{-1} - 1)\phi_{i-1/2,j} + V_i^{-2}\phi_{-(i-1/2),j} \quad (13)$$

Next, extend the definition $S_{i,j}$ in (11) to all integers and half-integers i and j such that $i+j$ is an integer. From (3) and (4) $S_{i,j} = 0$ for $-(i-1) \leq j \leq i-1$, and

$$S_{i,j} = \sum (\delta_{i,n} + h_{i,n})(\delta_{n,j} + k_{n,j}) = \sum \phi_{i,n}(\delta_{n,j} + k_{n,j})$$

$$+k_2(j+n)) = \phi_{1,j} + \phi_{1,j} * k_1(j) + \phi_{1,-j} * k_2(j) \quad (14)$$

where * denotes a convolution in j .

Since (13) is linear in functions of j , it may be convolved with $k_1(j)$. Adding (13) to the convolution of (13) with $k_1(j)$ and the convolution of the time-reversal of (13) with $k_2(j)$ and using (14) gives

$$S_{i+1/2,j} = S_{i,j+1/2} + S_{i,j-1/2} + (V_i^1 - 1)S_{i-1/2,j} + V_i^2 S_{i-1/2,j}. \quad (15)$$

Hence $S_{i,j}$ also satisfies the four-term recurrence (9).

B. New Split Schur Algorithm

Initialization: $S_{0,j} = k_{0,j}$; $S_{\pm 1/2,j+1/2} = k_{\pm 1/2,j+1/2}$

Computation of V_i^1, V_i^2 : Compute V_i^1 and V_i^2 from $S_{i,\pm}$ and $S_{i-1/2,\pm(i-1/2)}$ using (12). Similar equations are used to compute V_{-i}^1 and V_{-i}^2 .

Update $S_{i,j}, |j| \geq i$ using (15)

At this point the recursion is complete. The split Schur algorithm can be run in parallel with the split Levinson algorithm, supplying the potentials $V_{\pm i}^1$ and $V_{\pm i}^2$ while bypassing the "inner product" computation (11) (if (12) is still necessary), as suggested in [12] for the classical algorithm. Note $k_{0,m}$ and $k_{1/2,n+1/2}$ for integer m and half-integer $n+1/2$ uniquely determines $k_{i,j}$ for all i, j ; $i+j$ an integer, using (6).

If the original system (3) is a discretization of an integral equation, then $S_{i,j} \ll 1$ and the $\phi_{i,j}$ in (11) dominates the other terms if $i=j$. In this case the solution to (12) is simply $V_i^1 = S_{i-1/2,i-1} - S_{i,i}$ and $V_i^2 = S_{i-1/2,i-1} - S_{i,-i}$.

V. SOLUTION OF ARBITRARY TH SYSTEMS

The split algorithms above solve the systems (3) and (4); hence they also solve (1) with $S_{\pm i,\pm}$ defined as in (1). We now consider the general problem

$$\begin{bmatrix} 1+k_{-i,-i} & & k_{i,-i} \\ & \ddots & \\ k_{-i,i} & & 1+k_{i,i} \end{bmatrix} \begin{bmatrix} x_{-i} \\ \vdots \\ x_i \end{bmatrix} = \begin{bmatrix} b_{-i} \\ \vdots \\ b_i \end{bmatrix} \quad (16)$$

where the right side is now arbitrary.

Define $\{c_j, -i \leq j \leq i\}$ recursively as follows. Let $c_{\pm i}$ be the solution to the 2×2 system

$$\begin{bmatrix} S_{-i,-i} & S_{-i,i} \\ S_{i,-i} & S_{i,i} \end{bmatrix} \begin{bmatrix} c_{-i} \\ c_i \end{bmatrix} = \begin{bmatrix} b_{-i} - \sum_{n=-i+1}^{i-1} c_n S_{n,-i} \\ b_i - \sum_{n=-i+1}^{i-1} c_n S_{n,i} \end{bmatrix} \quad (17)$$

Then the solution to (16) is given by

$$x_j = \sum_{n=-i}^i c_n \phi_{n,j}, \quad -i \leq j \leq i. \quad (18)$$

These equations may be derived easily by taking linear combinations (weighted by the $c_{\pm i}$) of the columns of (1) for increasing i and equating to (16). Note how this relies on the split algorithms solving nested systems of equations as i increases.

The 2×2 systems (12) and (17) have unique solutions if the central submatrices of the system matrix (1) are nonsingular. To see this, suppose that the 2×2 system matrix in (12) and (17) is singular. Then the second column is a multiple (say m) of the first column, and the column vector $[1, \dots, (h_{-i,j} - mh_{i,j}), \dots, -m]^T$ solves the homogeneous system associated with (1), which is impossible as long as the system matrix in (1) is nonsingular.

VI. RELATION WITH PREVIOUS SPLIT ALGORITHMS

A. Relation to the Split Algorithms of [10]

To show how the new algorithms reduce to the split algorithms of [10], we first consider the class of TH matrices such that $k_{i,j} = k_{-i,-j}$. In terms of (2) both $k_{1,j}$ and $k_2(\cdot)$ are even functions; note that covariance functions of time-reversible random processes have this property. The set of centrosymmetric matrices (matrices that are both persymmetric $k_{i,j} = k_{-j,-i}$ and symmetric $k_{i,j} = k_{j,i}$) is a subset of this class. From (3) $h_{i,j} = h_{-i,-j}$, from (11) $S_{i,j} = S_{-i,-j}$, and from (12) $V_i^1 = V_{-i}^1$ and $V_i^2 = V_{-i}^2$. Hence the computations for $i < 0$ are all unnecessary.

We can go further. Defining

$$a_{i,j} = h_{i,j} + h_{i,-j}, \quad e_{i,j} = S_{i,j} + S_{i,-j}, \quad V_i = V_i^1 + V_i^2 \quad (19)$$

replacing j with $-j$ in (7) and (15) and adding to (7) and (15) respectively results in

$$\Delta a_{i,j} = V_i a_{i-1,j}; \quad \Delta e_{i,j} = V_i e_{i-1,j} \quad (20)$$

Adding the two equations of (12) allows V_i to be computed from $e_{i,j}$ by

$$V_i = (e_{i-1,i-1} - e_{i,i}) / (e_{i-1,-i-1} - e_{i,-i}) \quad (21)$$

From (3) and (19) $a_{i,j}$ is the solution to

$$k_{i,j} + k_{i,-j} = a_{i,j} + \sum_{n=-i+1}^{i-1} a_{i,n} k_{n,j}. \quad (22)$$

The solution to (22) can be recursively computed using the three-term recurrences (20), along with (21). These equations have virtually the same form as the split algorithms of [10], even though $k_{i,j}$ is not Toeplitz.

To see what is happening here, use (2) to rewrite the left side of (22) as

$$\begin{aligned} k_{i,j} + k_{i,-j} &= k_1(i-j) + k_2(i+j) + k_1(i+j) + k_2(i-j) \\ &= k(i-j) + k(i+j) \end{aligned} \quad (23)$$

where $k(i) = k_1(i) + k_2(i)$. From (19) $a_{i,j} = a_{i,-j}$, and the right side of (22) can be rewritten using this and (23), yielding

$$k(i-j) + k(i+j) = a_{i,j} + \sum_{n=0}^{i-1} a_{i,n} (k(n-j) + k(n+j)). \quad (24)$$

This is the symmetric Toeplitz system solved by the split algorithms of [10], after shifting from a one-sided to a two-sided interval. This shows how these algorithms are related to the algorithms of this paper. Note that the split algorithms of [10] propagate $a_{i,j}$, not $b_{i,j}$. $b_{i,j}$ must be computed from $a_{i,j}$ at the end.

VII. CONCLUSION

New fast algorithms have been derived for solving arbitrary TH systems of equations. The new algorithms can be viewed as analogues of the split Levinson and Schur algorithms of [10], but applicable to a more general problem. The split Levinson algorithm recursively computes the solution using a four-term recurrence, but requires a non-parallelizable computation (11) to compute the potentials. The split Schur algorithm computes the potentials using a similar four-term recurrence; using it in parallel with the split Levinson algorithm obviates (11) and allows the same processor architecture to be used for both algorithms.

ACKNOWLEDGMENT

This research was supported by the Air Force Office of Scientific Research under grant AFOSR-89-0017.

REFERENCES

1. A.-C. Lee, "A new autoregressive method for high-performance spectrum analysis," *J. Acoust. Soc. Am.*, vol. 86, no. 1, pp. 150-157, July 1989.
2. B. Friedlander and M. Morf, "Least-squares algorithms for adaptive linear-phase filtering," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-30, no. 3, pp. 381-389, June 1982.
3. S.M. Kay and S.L. Marple, "Spectrum analysis—a modern perspective," *Proc. IEEE*, vol. 69, no. 11, pp. 1380-1419, Nov. 1981, p. 1407.
4. W.R. Gragg and G.D. Johnson, "The Laurent-Pade table," in *Proc. IFIP Congr. 1974*, Amsterdam, the Netherlands, North Holland, 1974, pp. 632-637.
5. J. Casti, R. Kalaba, and S. Ueno, "Source functions for an isotropically scattering atmosphere bounded by a specular reflector," *J. Quant. Spectrosc. Radiat. Transfer*, vol. 10, pp. 1119-1128, 1970.
6. H.H. Kagiwada and R. Kalaba, *Integral Equations via Imbedding Methods*. Reading, Mass.: Addison-Wesley, 1974.
7. G.A. Merchant and T.W. Parks, "Efficient solution of a Toeplitz-plus-Hankel coefficient matrix system of equations," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-30, no. 1, pp. 40-44, Feb. 1982.
8. B. Friedlander and M. Morf, "Efficient inversion formulas for sums of products of Toeplitz and Hankel matrices," in *Proc. 18th Annual Allerton Conf. Commun. Comput.*, Oct. 8-10, 1980.
9. J.N. Tsitsiklis and B.C. Levy, "Integral equations and resolvents of Toeplitz plus Hankel kernels," Tech. Report #LIDS-P-1170, Laboratory for Information and Decision Systems, M.I.T., Cambridge, Mass., Dec. 1981.
10. P. Delsarte and Y. Genin, "On the splitting of classical algorithms in linear prediction theory," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-35, no. 5, pp. 645-653, May 1987.
11. A.E. Yagle, "Fast algorithms for estimation and signal processing: an inverse scattering framework," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, vol. ASSP-37, no. 6, pp. 957-959, June 1989.
12. S.Y. Kung and Y.H. Hu, "A highly concurrent algorithm and pipelined architecture for solving Toeplitz systems," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-31, no. 1, pp. 66-75, Feb. 1983.
13. P. Delsarte and Y. Genin, "Multichannel singular predictor polynomials," *IEEE Trans. Circuits and Systems*, vol. CAS-35, no. 2, pp. 190-200, Feb. 1988.
14. A.E. Yagle and W.-H. Fang, "Discrete fast algorithms for two-dimensional linear prediction on a polar raster," *Proc. 1990 IEEE Int'l Conf. on Acoust., Speech, Sig. Proc.*, Albuquerque, NM, April 3-6, 1990, pp. 2017-2020.
15. A.E. Yagle, "Generalized split Levinson, Schur, and lattice algorithms for three-dimensional random field estimation problems," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1780-1799, Dec. 1990.
16. A.E. Yagle, "New Analogues of Split Algorithms for Arbitrary Toeplitz-plus-Hankel Matrices," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-39, no. 11, Nov. 1991.

DISCRETE FAST ALGORITHMS FOR TWO-DIMENSIONAL LINEAR PREDICTION ON A POLAR RASTER

Wen-Hsien Fang and Andrew E. Yagle

Dept. of Electrical Engineering and Computer Science

The University of Michigan, Ann Arbor, Michigan 48109-2122

ABSTRACT

New discrete generalized split Levinson and Schur algorithms for the two-dimensional linear least-squares prediction problem on a polar raster are derived. The algorithms compute the prediction filter for estimating a random field at the edge of a disk, from noisy observations inside the disk. The covariance function of the random field is assumed to have a Toeplitz-plus-Hankel structure for both its radial part and its transverse part. This assumption can be shown to be closely related with some types of random fields, such as isotropic random fields. The algorithms generalize the split Levinson and Schur algorithms in two ways: (1) to two dimensions; and (2) to Toeplitz-plus-Hankel covariances.

I INTRODUCTION

The problem of computing linear least-squares estimates of two-dimensional random fields from noisy observations has many applications in image processing. In particular, the two-dimensional discrete linear prediction problem is a useful formulation of problems in smoothing and image coding and restoration[1].

If the random field: (1) is defined on a rectangular lattice of points; (2) is stationary; and (3) has quarter-plane or asymmetric half-plane causality, then the two-dimensional linear prediction problem may be solved using the multichannel Levinson algorithm [2,3,4].

However, in some medical imaging problems, and in spotlight synthetic aperture radar, data are collected on a polar raster of points, rather than on a rectangular lattice. Although such data can be interpolated onto a rectangular lattice, this is necessarily inexact; it also affects the covariance function. For restoring noisy images, image coding, etc., it is clearly desirable to develop analogues of the multichannel Levinson and Schur algorithms applicable to discrete random fields defined on a polar raster.

This paper develops these analogues. They generalize previous results in three ways: (1) the random field is defined on a polar raster; (2) the random field is not required to be stationary; rather, its covariance must have Toeplitz-plus-Hankel structure in both the radial and transverse directions; and (3) the quarter-plane or asymmetric half plane causality assumption is replaced by a more natural causality

in the radial direction only; the prediction filters estimate the random field at a given point using observations from all points of smaller radius. The algorithms are generalizations of the split algorithms [5,6].

This paper is organized as follows. In Section II, the two-dimensional analogues of the discrete split Levinson recurrence and split Schur recurrence for the linear prediction problem on a polar raster are derived. The derivation is based on the assumption that both the radial part and the transverse part of the covariance have Toeplitz-plus-Hankel structure. In Section III, an isotropic random field is shown to have a Toeplitz-plus-Hankel covariance, the overall complexity of the proposed algorithm is evaluated, and comparisons with the result of [7] are made. Section IV concludes with a summary and a discussion of how the results of this paper can be used to solve the general smoothing problem.

II DERIVATION OF THE RECURRENCE

A. Basic Problem

The problem considered is as follows. From noisy observations $\{y_{i,N}\}$ of a zero-mean real-valued discrete random field $\{x_{i,N}\}$ at the points (i, N) of a polar raster on a disk, compute the linear least-squares estimate of $x_{i,N}$ for all points on the edge of the disk. Here i is an integer radius from the origin, and N is the integer index of the argument (angle); if there are M points distributed on the circle of any radius, then (i, N) is the point at radius i and angle $2\pi N/M$.

The observations $\{y_{i,N}\}$ are related to the field $x_{i,N}$ by $y_{i,N} = x_{i,N} + v_{i,N}$, where $\{v_{i,N}\}$ is a zero-mean discrete white noise field with unit power, and $\{x_{i,N}\}$ and $\{v_{i,N}\}$ are uncorrelated. The covariance of $\{x_{i,N}\}$, $E[x_{i,N_1} x_{j,N_2}] = K(i, N_1; j, N_2)$, is assumed to be a non-negative definite function with Toeplitz-plus-Hankel structure in both arguments. The estimates of $x_{i,N}$ at the edge of the disk are computed from the observations $\{y_{i,N}\}$ using

$$\hat{x}_{i,N_1} = \sum_{j=0}^{i-1} \sum_{N_2=1}^M h(i, N_1; j, N_2) y_{j,N_2} \quad (1)$$

The optimal prediction filters $h(i, N_1; j, N_2)$ are computed by solving the two-dimensional discrete Wiener-Hopf equation

$$K(i, N_1; j, N_2) = h(i, N_1; j, N_2) + \sum_{n=-i-1}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) K(n, N_3; j, N_2) \quad (2)$$

for all $-(i-1) \leq j \leq i-1$ and $1 \leq N_1, N_2 \leq M$. The goal is to derive a fast algorithm for solving (2) when $K(i, N_1; j, N_2)$ has the Toeplitz-plus-Hankel structure shown (5) and (6) below.

We decompose the update procedure into two steps by introducing an *interpolated* (auxiliary) system. As shown in Figure 1, between every pair of points in the radial direction, we insert an auxiliary point. The covariance function $K(i, N_1; j, N_2)$ is interpolated at these auxiliary points such that the block Toeplitz-plus-Hankel structure (see (5), (6)) is maintained. Then the prediction filter can be defined at the interpolated points as the solution to the interpolated system, which has the form of (2) but is specified on the interpolated points.

B. Derivation of the Levinson-Like Recurrence

Define the discrete wave operators Δ_r and Δ_θ by

$$\Delta_r f(i, N_1; j, N_2) = f(i + \frac{1}{2}, N_1; j, N_2) + f(i - \frac{1}{2}, N_1; j, N_2) - f(i, N_1; j + \frac{1}{2}, N_2) - f(i, N_1; j - \frac{1}{2}, N_2) \quad (3)$$

$$\Delta_\theta f(i, N_1; j, N_2) = f(i - \frac{1}{2}, ((N_1 + 1)); j, ((N_2))) + f(i - \frac{1}{2}, ((N_1 - 1)); j, ((N_2))) - f(i - \frac{1}{2}, ((N_1)); j, ((N_2 + 1))) - f(i - \frac{1}{2}, ((N_1)); j, ((N_2 - 1))) \quad (4)$$

where Δ_r and Δ_θ can be regarded as discrete versions of the continuous operators $(\frac{\partial^2}{\partial^2 r_1} - \frac{\partial^2}{\partial^2 r_2})$ and $(\frac{\partial^2}{\partial^2 \theta_1} - \frac{\partial^2}{\partial^2 \theta_2})$ for the radial part and transverse part, respectively, and $((\cdot))$ means a mod M operation.

We assume that the covariance function has the block Toeplitz-plus-Hankel structure

$$\Delta_\theta K(i, N_1; j, N_2) = 0 \quad (5)$$

$$\Delta_r K(i, N_1; j, N_2) = 0 \quad (6)$$

Some examples satisfying (5) and (6) can be found in [8].

Applying the Laplacian operator $\Delta = \Delta_r + \Delta_\theta$ to the equation (2), we have after some algebra [8]

$$\begin{aligned} h(i + \frac{1}{2}, N_1; j, N_2) &= h(i, N_1; j + \frac{1}{2}, N_2) + h(i, N_1; j - \frac{1}{2}, N_2) \\ &- h(i - \frac{1}{2}, N_1; j, N_2) + h(i - \frac{1}{2}, N_1; j, N_2 + 1) + h(i - \frac{1}{2}, N_1; j, N_2 - 1) \\ &- h(i - \frac{1}{2}, N_1 + 1; j, N_2) - h(i - \frac{1}{2}, N_1 - 1; j, N_2) + \sum_{N_3=1}^M \\ [V_i^+(N_1, N_3) h(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3) h(-(i - \frac{1}{2}), N_3; j, N_2)] \end{aligned} \quad (7)$$

for all $-(i - \frac{3}{2}) \leq j \leq (i - \frac{3}{2})$ and $1 \leq N_1, N_2 \leq M$. Here we have defined the potentials

$$V_i^+(N_1, N_2) = -[h(i + \frac{1}{2}, N_1; i - \frac{1}{2}, N_2) - h(i, N_1; i - 1, N_2)] \quad (8)$$

$$V_i^-(N_1, N_2) = -[h(i + \frac{1}{2}, N_1; -i + \frac{1}{2}, N_2) - h(i, N_1; -i + 1, N_2)] \quad (9)$$

Equation (7) is the basic recurrence that is the heart of the Levinson-like algorithm. The left side is the difference of two two-dimensional discrete Laplacian operators, analogous to the difference of one-dimensional discrete Laplacian operators appearing in the split algorithms of [5]. The right side generalizes the three-term recurrence in [5] to a multi-term recurrence; this is analogous to the matrix recurrence in [6]. However, it is applicable to non-symmetric block Toeplitz-plus-Hankel systems (see [8]).

When i is an integer and j is a half-integer, equation (7) will update h from the real points to the interpolated points. When i is a half-integer and j is an integer, equation (7) will update h from the interpolated points to the real points.

C. Derivation of the Schur-Like Recurrence

We still need to calculate the potentials $V_i^+(N_1, N_2)$ and $V_i^-(N_1, N_2)$ at the beginning of every update so that we can use the recursive formula (7). Since an inner product is a bottle neck in a parallel processing environment, we overcome this difficulty by introducing the *Schur variables* (defined at integer and half-integer points)

$$s(i, N_1; j, N_2) \equiv \delta_{i, N_1; j, N_2} + K(i, N_1; j, N_2) - h(i, N_1; j, N_2) - \sum_{n=-i-1}^{i-1} \sum_{N_3=1}^M h(i, N_1; n, N_3) K(n, N_3; j, N_2) \quad (10)$$

where $\delta_{i, N_1; j, N_2} = 0$ unless $i = j$ and $N_1 = N_2$, in which case it is unity.

Since the Schur variables are the linear combinations of the prediction *error* filters $\delta_{i, N_1; j, N_2} - h(i, N_1; j, N_2)$, equations (7)-(8) show that $s(i, N_1; j, N_2)$ satisfies the recurrence (7), but now for *all* j :

$$\begin{aligned} s(i + \frac{1}{2}, N_1; j, N_2) &= s(i, N_1; j + \frac{1}{2}, N_2) + s(i, N_1; j - \frac{1}{2}, N_2) \\ &- s(i - \frac{1}{2}, N_1; j, N_2) + s(i - \frac{1}{2}, N_1; j, N_2 + 1) + s(i - \frac{1}{2}, N_1; j, N_2 - 1) \\ &- s(i - \frac{1}{2}, N_1 + 1; j, N_2) - s(i - \frac{1}{2}, N_1 - 1; j, N_2) + \sum_{N_3=1}^M \\ [V_i^+(N_1, N_3) s(i - \frac{1}{2}, N_3; j, N_2) + V_i^-(N_1, N_3) s(-(i - \frac{1}{2}), N_3; j, N_2)] \end{aligned} \quad (11)$$

Equation (11) is the basic recurrence for the Schur-like algorithm; for $-(i-1) \leq j \leq (i-1)$, $s(i, N_1; j, N_2) = 0$ by (2).

Setting $j = (i - \frac{1}{2})$ and $-(i - \frac{1}{2})$ in (11) respectively, we can solve for V_i^+ and V_i^- in closed form as [8]:

$$\begin{aligned} \hat{V}^+ &= (\mathbf{X} - \mathbf{Y}(\hat{\mathbf{S}}^{--})^{-1} \hat{\mathbf{S}}^{--})(\hat{\mathbf{S}}^{++} - \hat{\mathbf{S}}^{+-}(\hat{\mathbf{S}}^{--})^{-1} \hat{\mathbf{S}}^{--})^{-1} \mathbf{I}_2 \\ \hat{V}^- &= (\mathbf{Y} - \mathbf{X}(\hat{\mathbf{S}}^{++})^{-1} \hat{\mathbf{S}}^{++})(\hat{\mathbf{S}}^{--} - \hat{\mathbf{S}}^{+-}(\hat{\mathbf{S}}^{++})^{-1} \hat{\mathbf{S}}^{--})^{-1} \mathbf{I}_2 \end{aligned}$$

where we have defined the $M \times M$ matrices

$$[\tilde{V}^+]_{N_1, N_2} = V_i^+(N_1, N_2); [\tilde{V}^-]_{N_1, N_2} = V_i^-(N_1, N_2) \quad (14)$$

$$[\tilde{S}^\pm]_{N_1, N_2} = s(\pm(i - \frac{1}{2}), N_1; \pm(i - \frac{1}{2}), N_2) \quad (15)$$

$$\begin{aligned} X_{N_1, N_2} &= s(i - \frac{1}{2}, N_1; i - \frac{1}{2}, N_2) - s(i, N_1; i, N_2) \\ &\quad + \Delta_\theta s(i, N_1; i - \frac{1}{2}, N_2) \end{aligned} \quad (16)$$

$$\begin{aligned} Y_{N_1, N_2} &= s(i - \frac{1}{2}, N_1; -(i - \frac{1}{2}), N_2) - s(i, N_1; -i, N_2) \\ &\quad + \Delta_\theta s(i, N_1; -(i - \frac{1}{2}), N_2) \end{aligned} \quad (17)$$

D. Summary of Overall Procedure

The overall procedure can be summarized as follows. Let I_{\max} be the largest radius (maximum radial prediction order). Ther for all $1 \leq N_1, N_2 \leq M$:

1. Initialization

Compute $h(\pm\frac{1}{2}, N_1; 0, N_2)$, $h(\pm 1, N_1; 0, N_2)$ using (2).

Compute $s(\pm\frac{1}{2}, N_1; j, N_2)$, $s(\pm 1, N_1; j, N_2)$ using (10) for all $j = \pm 1, \dots, \pm 2I_{\max}$.

2. Propagation of Split Schur-Like Algorithm

A. Compute the potentials $V_i^+(N_1, N_2)$ and $V_i^-(N_1, N_2)$ using (12) and (13);

B. Update the Schur variables using (11) for $j = \pm(i + \frac{1}{2}), \dots, \pm 2I_{\max}$.

3. Propagation of Split Levinson-Like Recurrence

A. Propagate the Boundary Points:

$$h(i + \frac{1}{2}, N_1; i - \frac{1}{2}, N_2) = h(i, N_1; i - 1, N_2) - V_i^+(N_1, N_2) \quad (18)$$

$$h(i + \frac{1}{2}, N_1; -i + \frac{1}{2}, N_2) = h(i, N_1; -i + 1, N_2) - V_i^-(N_1, N_2) \quad (19)$$

B. Propagate Non-Boundary Points:

Update $h(i, N_1; j, N_2)$ using equation (7) for $j = -(i - \frac{1}{2})$ to $j = (i - \frac{1}{2})$.

4. Repeat steps 2 and 3 from $i = 1$ to I_{\max} with increment $\frac{1}{2}$.

Note that the above generalized Levinson and Schur recurrences (7) and (11) are highly parallel, and perform the same type of in-place computation. This allows a highly parallel and pipelined architecture to be developed for this algorithm.

III DISCUSSION

A. Isotropic Random Field

For an isotropic random field, the covariance is a function of distance only, i.e., if x and y are two arbitrary points in the plane, then $K(x, y) = K(|x - y|)$. Consider the special case

of a isotropic random field with covariance $K(x, y) = \rho^{|x-y|^2}$, which is often used in image modeling. In polar coordinates on a discrete polar raster, and if $\rho \approx 1$, this covariance function can be represented as

$$\begin{aligned} K(i, N_1; j, N_2) &= \rho^{i^2+j^2-2ij\cos(2\pi(N_1-N_2)/M)} \\ &= \rho^{\frac{1}{2}[(i+j)^2+(i-j)^2]-[(i+j)^2-(i-j)^2]\cos(2\pi(N_1-N_2)/M)} \approx 1 + \frac{1}{2} \\ &\quad ([(i+j)^2+(i-j)^2] - [(i+j)^2-(i-j)^2] \cos(2\pi(N_1-N_2)/N) \ln \rho(20) \end{aligned}$$

Note that the exponent has the Toeplitz-plus-Hankel structure required by (5) and (6), and that it is not merely block-Toeplitz; hence the multichannel Levinson algorithm is not applicable. If $\rho \approx 1$, the entire covariance satisfies (5) and (6). Indeed, any slowly-changing function of distance satisfies (5) and (6).

B. Computational Complexity

We determine the number of multiplications/divisions (MADs) needed to solve (4) up to order $i = I_{\max}$. The initialization of the Levinson-like recurrences requires $2M \times M$ matrix inversions and $4M \times M$ matrix multiplications, or $2(\frac{M^3}{3} + \frac{M^2}{2}) + 4M^3$ MADs. The initialization of the Schur-like recurrences requires $8I_{\max}M \times M$ matrix multiplications, or $8I_{\max}M^3$ MADs. Each Schur-like recursion update of $s(i, N_1; j, N_2)$ from i to $i + \frac{1}{2}$ requires $16(I_{\max} - i)M^2$ MADs. Computation of the potentials requires $4M \times M$ matrix inversions and $6M \times M$ matrix multiplications. Finally, updating $h(i, N_1; j, N_2)$ from i to $i + \frac{1}{2}$ in the Levinson-like recurrence requires $4(2i + 1)M^2$ MADs. The total number of multiplications needed to solve (2) up to $i = I_{\max}$ is equal to [8]

$$24I_{\max}^2M^2 + I_{\max}(\frac{68M^3}{3} + 4M^2) + (\frac{14M^3}{3} + M^2) = O(I_{\max}^2M^2) \quad (21)$$

For large I_{\max} , this is much less than the number of MADs required for the solution of (4) by Gaussian elimination, which would require $\frac{(2I_{\max}M)^3}{3} + \frac{(2I_{\max}M)^2}{2} = O(I_{\max}^3M^3)$ multiplications. In addition, as shown in the above procedures, this procedure is highly parallelizable. Therefore, the overall reduction in time complexity would be even more significant using vector/parallel processors.

C. Relations with Continuous Algorithms

It is instructive to examine the continuous-parameter limits of some of the equations of this paper. Let the intervals between points be δ_r in the radial direction and $\delta_\theta = \frac{2\pi}{M}$ radians in the transverse direction. Introducing a radial weighting factor, and taking limits as δ_r and δ_θ go to zero result in the following transformations:

1. The discrete Wiener-Hopf equation (2) becomes the Wiener-Hopf integral equation;
2. δ_{i, N_1, N_2} becomes a continuous two-dimensional impulse function, dominating the other terms in the definition (10) of the Schur variables. The recursion (11)

now propagates the non-impulsive part of the Schur variables, so that (12) and (13) may be replaced with $\bar{V}^+ \approx X$ and $\bar{V}^- \approx Y$. Compare this to

$$V(x, \theta_1; \theta_2) = -\left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y}\right)s(x, \theta_1; y = x, \theta_2) \quad (22)$$

where x and y are continuous radii and θ_1 and θ_2 are continuous angles. Equation (22) has the form of (4-17b) of [7]. Similarly, the continuous version of (13) has the form of (4-2) of [7]. Equation (7), with its difference of discrete two-dimensional Laplacian operators on the left side, is clearly analogous to $(\Delta_x = \text{Laplacian with respect to } x)$

$$(\Delta_x - \Delta_y)h(x, \theta_1; y, \theta_2) = \int_0^{2\pi} V(x, \theta_1; \theta_3)h(x, \theta_3; y, \theta_2) d\theta_3 \quad (23)$$

which is the two-dimensional form of (4-1) of [7]. However, (23) is NOT the continuous limit of (15) with radial weighting, since $\frac{1}{\sqrt{x}} \frac{d^2}{dx^2}(\sqrt{x}f(x)) = (\frac{d^2}{dx^2} + \frac{1}{x} \frac{d}{dx} - \frac{1}{4x^2})f(x)$, which is not the radial part of the 2-D Laplacian. On the other hand, $\frac{1}{x} \frac{d^2}{dx^2}(xf(x)) = (\frac{d^2}{dx^2} + \frac{2}{x} \frac{d}{dx})f(x)$, which is the radial part of the 3-D Laplacian. This shows that the results of [7], derived for the continuous 3-D case, do not apply *exactly* to the 2-D case (as do the results of this paper);

3. The algorithms of this paper require the differences of the radial parts and transverse parts of the Laplacian of the covariance to be *separately* zero: (5) and (6) must be *separately* zero. However, in the continuous limit, we have $h(i, N_1; n, N_3) \approx h(i - \frac{1}{2}, N_1; n, N_3)$; then it suffices for the sum $(\Delta_r + \Delta_\theta)K(i, N_1; j, N_3) = 0$, rather than (5) and (6) separately. This agrees with the requirement $(\Delta_x - \Delta_y)K(x, y) = 0$ for the algorithms in [7].

We can draw some important conclusions from these observations. If the algorithms of this paper are being used to solve a discretized Wiener-Hopf equation, then

- (a) In (12) and (13) the impulses lead to diagonally-dominant systems, so that (12) and (13) may be replaced with the approximations $\bar{V}^+ \approx X$ and $\bar{V}^- \approx Y$. Therefore, the overall complexity will be furtherly reduced by avoiding the matrix inversions in (12) and (13);
- (b) By the chain rule, *any* continuous function of the distance between two points will satisfy (5) and (6), since the square of the distance itself does. Hence the algorithms may be used for any isotropic random field. Note in particular that (20) becomes

$$K(i, N_1; j, N_2) = \rho^{\delta^2(i^2 + j^2 - 2ij \cos(N_1 - N_2)\delta_\theta)} \quad (24)$$

and $\rho^{\delta^2} \rightarrow 1$ as $\delta_r \rightarrow 0$;

- (c) Conditions (5) and (6) may be replaced with the more general condition $(\Delta_r + \Delta_\theta)K(i, N_1; j, N_3) = 0$.

Numerical studies have shown that approximation (2) give very good results for $\delta_r \approx 0.001$, but discretization is much more sensitive to non-infinitesimal δ_r .

IV CONCLUSION

New fast algorithms for solving the discrete 2-D Wiener-Hopf equation on a polar raster when the covariance function has block Toeplitz-plus-Hankel structure have been derived. Since we have performed explicitly discrete derivations, instead of just discretizing the continuous versions, the algorithms work regardless of the number of points used. If adjacent points are close enough, then the algorithm would reduce to the continuous case [7].

The *smoothing* filter for estimating the points *inside* the disk can be computed from the prediction filters using a generalized discrete Bellman-Siebert-Krein identity. The overall complexity is reduced compared with Gaussian elimination [9].

Unresolved issues include mapping of this algorithm into optimal array processor architectures, the numerical stability of the algorithm, and practical applications of this algorithm in problems such as image restoration and coding.

ACKNOWLEDGMENT

The work of both authors was supported by the Air Force Office of Scientific Research under grant # AFOSR-89-0017.

- [1] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [2] J.H. Justice, *A Levinson-Type Algorithm for Two-Dimensional Wiener Filtering using Bivariate Szego Polynomials*, Proc. IEEE, vol. 65, no.6, pp. 882-886, June 1977.
- [3] C.W. Therrien, *Relations between 2-D and Multichannel Linear Prediction*, IEEE Trans. ASSP, vol. ASSP-29, no.6, pp. 454-456, June 1981.
- [4] T. Marzetta, *A Linear Prediction Approach to Two-Dimensional Spectral Factorization and Spectral Estimation*, Ph.D. Thesis, Dept. of EECS, M.I.T., Cambridge, MA, 1978.
- [5] P. Delsarte, Y. Genin, *On the Splitting of Classical Algorithms in Linear Prediction Theory*, IEEE Trans. ASSP, vol. ASSP-35, no.5, pp. 645-653, May 1987.
- [6] P. Delsarte, Y. Genin, *Multichannel Singular Predictor Polynomials*, IEEE Trans. Circuit and Systems, vol. CAS-35, no.2, pp. 190-200, Feb. 1988.
- [7] A.E. Yagle, *Analogues Split Levinson, Schur, and Lattice Algorithms for Three-Dimensional Random Field Estimation Problem*, to appear in SIAM J. Appl. Math.
- [8] W.-H. Fang, A.E. Yagle, *Discrete fast algorithms for two-dimensional linear prediction on a polar raster*, submitted to IEEE Trans. ASSP, Oct. 1989.
- [9] W.-H. Fang, A.E. Yagle, *Fast Algorithms for Linear Least-Squares Smoothing Problems in One and Two Dimensions Using Generalized Discrete Bellman-Siebert-Krein Resolvent Identities*, submitted to IEEE Trans. ASSP, Oct. 1989.

Session TE1 – Multidimensional Signal Processing Algorithms and Techniques

Generalized Levinson and Schur Algorithms for
Multi-Dimensional Random Field Estimation Problems

Andrew E. Yagle

Dept. of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2122

Fast algorithms for computing the linear least-squares estimate of a multi-dimensional random field from noisy observations inside a circle (2-D) or sphere (3-D) are derived. The double Radon transform of the random field covariance is assumed to have a Toeplitz-plus-Hankel structure; this is equivalent to the multi-dimensional spatial displacement property $(\Delta_x - \Delta_y)k(x, y) = 0$. Note that this only reduces the number of degrees of freedom by one; homogeneous and isotropic random fields are included as special cases. The algorithms exploit this structure to reduce the amount of computation needed to solve the multi-dimensional Wiener-Hopf equation

$$k(x, y) = h(x, y) + \int h(x, z)k(z, y)dz, \quad |y| < |x|, \quad x, y, z \in R^3$$

The algorithms can be viewed as generalized split Levinson and Schur algorithms, since they exploit this structure in the same way that their one-dimensional counterparts exploit the Toeplitz structure of the covariance of a stationary random process. The algorithms are easily parallelizable, and they are recursive in increasing radius of the hypersphere of observations. They have the form

$$(\Delta_x - \Delta_y)h(x, y) = \int V(x, e)h(|x|e, y)de, \quad \|e\| = 1, \quad x, y \in R^3$$

where $V(x, e)$ characterizes the filters $h(z, y)$ for $|y| < |z| < |x|$ much as the reflection coefficients characterize the 1-D prediction filters of all orders. The discrete forms of the problem and the algorithm are shown to be simply the obvious discretizations of the equations given here.

It is important to note that these algorithms do NOT assume quarter-plane or asymmetric half-plane support for the filter, as do previous "2-D" Levinson algorithms that are really multichannel 1-D algorithms. The new algorithms are true multi-dimensional algorithms that do not attempt to reduce dimensionality, but only take advantage of an assumed structure of the covariance function.

An earlier version of this work was presented at the ICASSP in New York. The new material presented here includes:

1. The discrete form of the problem, and the discrete algorithm solving it;
2. Numerical results on the performance of the algorithm;
3. A procedure for estimating a covariance of the desired form from a sample function of a random field (i.e., a multi-dimensional "Toeplitzation plus Hankelization")